Robot experiment with PIC microcontroller

based-on Robo-PICA robot kit

INNOVATIVE EXPERIMENT

Contents

Chapter 1

Part list of Robo-PICA and Introduce software tool

1.1 Robo-PICA part list

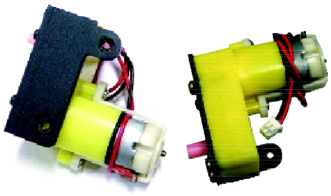
There is 3 groups as :

1.1.1 Mechanical parts

1.1.2 Electronic parts

1.1.3 Optional parts

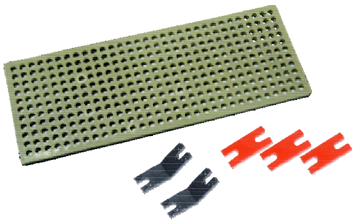
1.1.1 Mechanical parts



Motor Gearbox – Uses a 4.5-9V and 180 mA DC motor with a ratio of 48:1; torque 4kg/cm; comes with support plates.

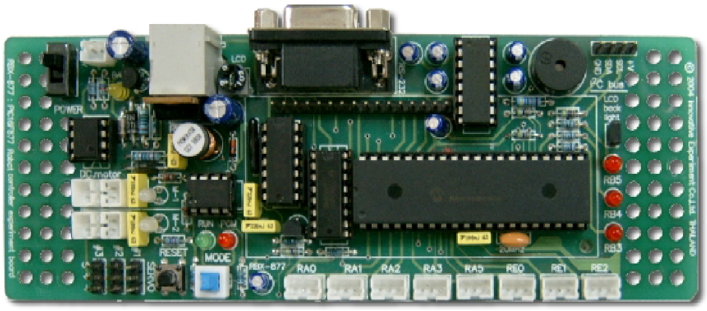
Track wheel set - includes 3-length of Track wheel, many support wheels and sprockets, axels and shaft base

Many sizes of Screw and Nut (4 of 3x6mm., 30 of 3x10mm., 4 of 3x15mm., 4 of 3x25 mm., 4 of 3x35mm. Screw and 30 of 3mm. nuts), 2 of Flat head screws, **Set of Spacers** (4 of 5mm., 4 of 10mm., 4 of 15mm. and 4 of 25 mm.)

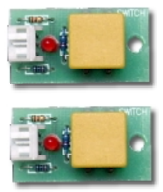


The Plate set and **3-types of the color-mixed Plastic Joiner** (20 of Strangth Joiner, 20 of Right-angle Joiner and 20 of Obtuse Joiner)

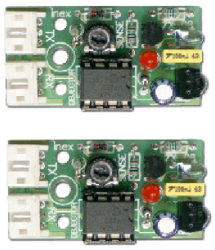
1.1.2 Electronic parts



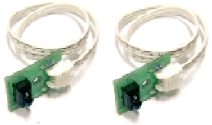
RBX-877 PIC16F877 Robot Experiment Board



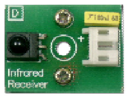
ZX-01 Switch input (x2)



ZX-08 Infrared Objector (x2)

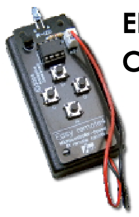


ZX-03 Infrared Reflector (x2)

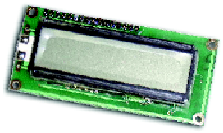


ZX-05 38kHz Infrared Receiver

AA size 4-pieces Battery Holder with power cable

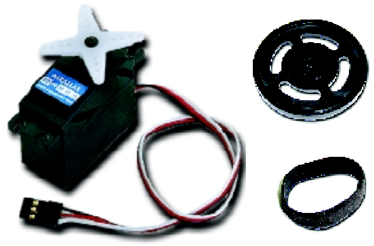


ER-4 Infrared Remote Control

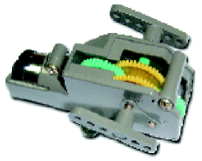


16-Characters 2-lines LCD module with Back-light assemblied female connector

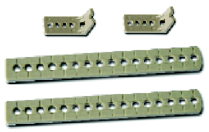
1.1.3 Optional parts (not included in the standard kit, supply separated)



4.8-6V RC servo motor with **Circular wheel and Tire**

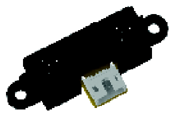


4-Speed motor gearbox



Arm set

Four of AA size Rechargeable batteries (1800mAH or higher recommended)



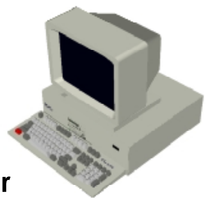
GP2D120 4 to 30cm. Infrared Distance sensor

1.2 Tools for making the robot kit

Cutter plier

A sharp-tipped hobby knife or Handy Cutter

Philips Screwdriver



Computer
Install Windows98SE or higher and has both RS-232 serial port and Parallel port

1.3 Software development tools for Robot programming

In Robo-PICA kit use PIC microcontroller, PIC16F877 then builders can write the controlled program in assembly, BASIC and C language. Only BASIC and C program must use compiler software.

However in this kit select the BASIC program in all examples. Because BASIC program is easy to learn and suitable for beginners. The most popular of BASIC compiler for PIC microcontroller is PICBASIC PRO compiler from Microengineering Labs (Melabs : www.melabs.com). Exactly, the Robo-PICA robot kit use this compiler too.

The demo version of PICBASIC PRO compiler is selected for this robot kit. Builders who need to develop the advance program, please purchase the full version from Melabs at their webiste. The demo version of PICBASIC PRO can download from *http://www.melabs.com/pbpdemo.htm*. However in the Robo-PICA robot kit contains this software in a bundled CD-ROM.

Builders must download the PICBASIC PRO manual from Melabs for programming reference. In this manual does not describe the instruction sets of all.

1.3.1 PicBasic Pro Compiler (Demo version)

This demo version of PICBASIC PRO Compiler supports a limited number of PIC microcontroller and is limited to 31 lines of source code (comments and whitespace are not counted) and cannot include another file by `INCLUDE` directive.

Builders can view the PICBASIC PRO Compiler manual from Melabs developer's resources page.

Demo version PICmicro MCU support:

PIC16F627(A), PIC16F628(A), PIC16F84(A), PIC16F870, PIC16F871, PIC16F872, PIC16F873(A), PIC16F874(A), PIC16F876(A) and PIC16F877(A)

The PICBASIC PRO compiler's operation is compile the BASIC language program in .bas file to machine code in .hex file. It has 2 steps as :

1. Compile the BASIC language programs to assembly program. The result file is .asm file.

2. Assembler the assembly program to the result file in .hex file.

1.3.1.1 Reserve words of PICBASIC PRO compiler

Reserve words mean the words or message that reserved by compiler. Programmer cannot use all words for varaible name definition and any label in their program. All reserved word are sequenced following alphabet numerical as :

A

ABS ADCIN AND ANDNOT ASM AUXIO

B

BANK0 BANK1 BANK2 BANK3 BANK4 BANK5 BANK6 BANK7
BANK8 BANK9 BANK10 BANK11 BANK12 BANK13 BANK14 BANK15
BANKA BIN BIN1 BIN2 BIN3 BIN4 BIN5 BIN6
BIN7 BIN8 BIN9 BIN10 BIN11 BIN12 BIN13 BIN14
BIN15 BIN16 BIT BRANCH BRANCHL BUTTON BYTE

C

CALL CASE CLEAR CLEARWDT CON COS COUNT

D

DATA DCD DEBUG DEBUGIN DEC DEC1 DEC2 DEC3
DEC4 DEC5 DEFINE DIG DISABLE DIV32 DTMFOUT

E

EEPROM ELSE ENABLE END ENDASM ENDIF ERASECODE EXT

F

FOR FREQOUT

G

GET GOSUB GOTO

H

HEX HEX1 HEX2 HEX3 HEX4 HEX5 HIGH HPWM
HSERIN HSEROUT

I

I2CREAD I2CWRITE IBIN IBIN1 IBIN2 IBIN3 IBIN4 IBIN5
IBIN6 IBIN7 IBIN8 IBIN9 IBIN10 IBIN11 IBIN12 IBIN13
IBIN14 IBIN15 IBIN16 IDEC IDEC1 IDEC2 IDEC3 IDEC4
IDEC5 IF IHEX IHEX1 IHEX2 IHEX3 IHEX4 IHEX5
INCLUDE INPUT INTERRUPT IS ISBIN ISBIN1 ISBIN2 ISBIN3
ISBIN4 ISBIN5 ISBIN6 ISBIN7 ISBIN8 ISBIN9 ISBIN10 ISBIN11
ISBIN12 ISBIN13 ISBIN14 ISBIN15 ISBIN16 ISDEC ISDEC1 ISDEC2
ISDEC3 ISDEC4 ISDEC5 ISHEX ISHEX1 ISHEX2 ISHEX3 ISHEX4
ISHEX5

L

LCDIN LCDOUT LET LIBRARY LOOKDOWN LOOKDOWN2 LOOKUP LOOKUP2
LOW

M

```
MAX      MIN      MOD
```

N

```
NAP      NCD      NEXT     NOT
```

O

```
OFF      ON       OR       ORNOT    OUTPUT   OWIN     OWOUT
```

P

```
PAUSE    PAUSEUS  PEEK     PEEKCODE POKE     POKECODE POLLIN   POLLMODE
POLLOUT  POLLRUN  POLLWAIT POT      PULSIN   PULSOUT  PUT      PWM
```

R

```
RANDOM   RCTIME   READ     READCODE REM      REP      RESUME   RETURN
REV      REVERSE
```

S

```
SBIN     SBIN1    SBIN2    SBIN3    SBIN4    SBIN5    SBIN6    SBIN7
SBIN8    SBIN9    SBIN10   SBIN11   SBIN12   SBIN13   SBIN14   SBIN15
SBIN16   SDEC     SDEC1    SDEC2    SDEC3    SDEC4    SDEC5    SELECT
SERIN    SERIN2   SEROUT   SEROUT2  SHEX     SHEX1    SHEX2    SHEX3
SHEX4    SHEX5    SHIFTIN  SHIFTOUT SIN      SKIP     SLEEP    SOUND
SQR      STEP     STOP     STR      SWAP     SYMBOL   SYSTEM
```

T

```
THEN     TO       TOGGLE
```

U

```
USBIN    USBINIT  USBOUT
```

V

```
VAR
```

W

```
WAIT     WAITSTR  WEND     WHILE    WORD     WRITE    WRITECODE
```

X

```
XIN      XOR      XORNOT   XOUT
```

1.3.1.2 Installation the PICBASIC PRO compiler demo version

If download PICBASIC PRO compiler demo version success, builders can install by their own. Download and run the install file. This will install the PBP demo and the Windows interface, MicroCode Studio. To get started programming, launch MicroCode Studio from your start menu.

The PICBASIC PRO Manual can be found in the MicroCode Studio help file or you can view the online manual at www.melabs.com

1.3.1.3 Using PICBASIC PRO compiler on Window Operating System

The PICBASIC PRO compiler is a compiler that operate on DOS OS and can run in command line prompt. However some third party company develop the Windows IDE for PICBASIC PRO compiler. The most popular are **Code Designer from CSMicro Systems** and **Microcode Studio from Mecanique**. In this mnual suggess to use the Microcode Studio. Because it's free software and can works with PICBASIC PRO all full and demo version. Addition the Microcode Studio GUI is very friendly user and easy to use.

The installation of PICBASIC PRO compiler onto Microcode Studio very comfortable. The Microcode Studio can find the folder that contains PICBASIC PRO compiler automatically.

1.3.2 The Microcode Studio

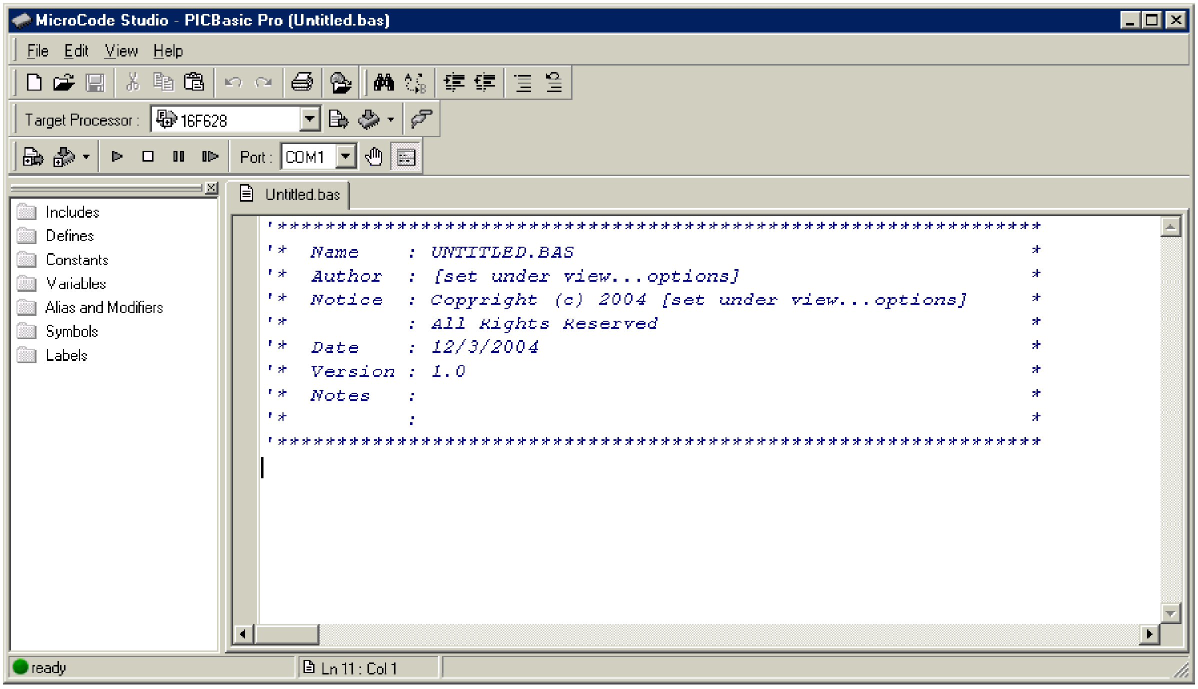
Microcode Studio is developed by **Mecanique** from UK. Free download the latest version at www.mecanique.com. MicroCode Studio is a powerful visual Integrated Development Environment (IDE) with In Circuit Debugging (ICD) capability designed specifically for MicroEngineering Labs PICBasic PRO compiler.

It's easy to set up your compiler, assembler and programmer options or you can let MicroCode Studio do it for you with its built in auto search feature. Compilation and assembler errors can easily be identified and corrected using the error results window. MicroCode Studio even comes with a serial communications window.

1.3.2.1 Installation Microcode Studio

Buider can install the Microcode Studio software from mcstudio.exe file that contained in Robo-PICA CD-ROM. Enter folder Tools → MicroCode Studio 2.1.0.1 (the number of version could be change depend on the lateset version). Double-click the mcstudio.exe file for starting the installation. Click OK button for acception the installastion process until the installation comlete.

To run the Microcode Studio, select to Start → Program → Microcode Studio → Microcode Studio. The main window of Microcode Studio will appear, see figure 1-1



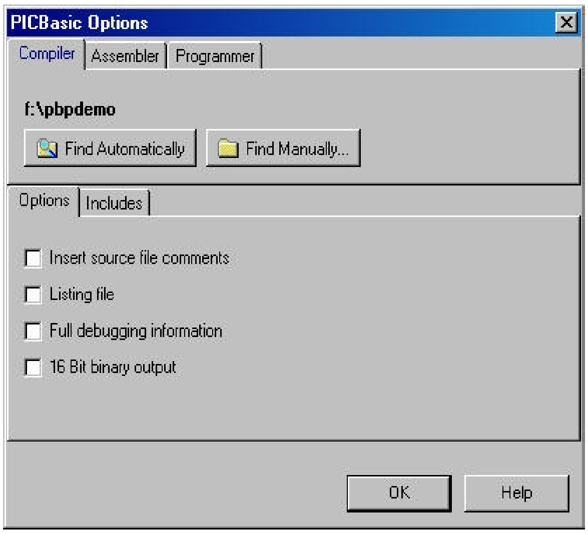
```
'****************************************************************
'*  Name    : UNTITLED.BAS                                      *
'*  Author  : [set under view...options]                        *
'*  Notice  : Copyright (c) 2004 [set under view...options]     *
'*          : All Rights Reserved                               *
'*  Date    : 12/3/2004                                         *
'*  Version : 1.0                                               *
'*  Notes   :                                                   *
'*          :                                                   *
'****************************************************************
```

Figure 1-1 Main window of the Microcode Studio

1.3.2.2 Adding PICBASIC PRO compiler

MicroCode Studio will automatically scan for the PICBasic PRO demo compiler when started for the first time. If you have another microEngineering Labs compiler installed (for example, the standard version) you will need to manually point MicroCode Studio to the correct installation folder.

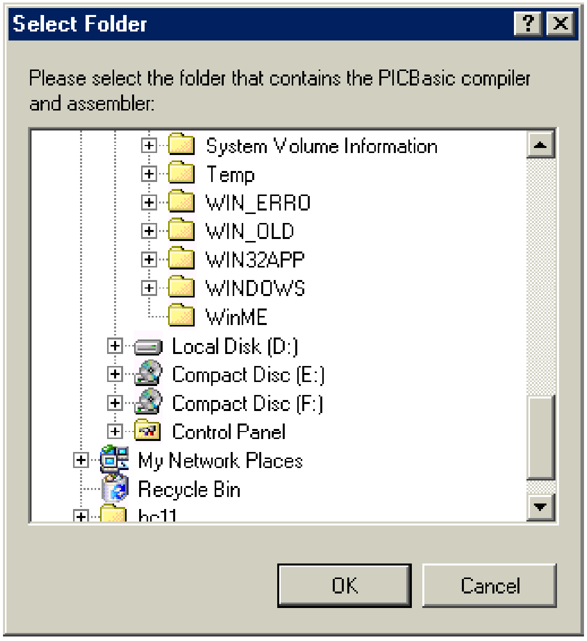
To do this, select VIEW → PIC BASIC OPTIONS from the main menu. Make sure the 'Compiler' tab is selected following the figure 1-2. Next, press the 'Find Manually...' button and select the folder containing the demo version of the compiler following the figure 1-3. After that click **OK** button to confirmation.



f:\pbpdemo

□ Insert source file comments
□ Listing file
□ Full debugging information
□ 16 Bit binary output

Figure1-2 PICBasic Option window, use to select the folder that install PICBASIC PRO compiler



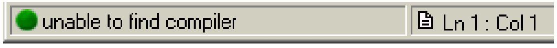
Please select the folder that contains the PICBasic compiler and assembler:

Figure 1-3 Select folder window of PICBASIC PRO compiler in case find manually

---

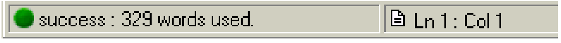
Note : *User can check the correct folder of compiler by open the example program, press F9 key for compiling program. Observe the status bar at the bottom of wmain window.*

*(A) If **unable to find compiler** message appears. It means the folder is not correct. Need to change.*



unable to find compiler

*(B) If **Success : xx words used** message appears. It means everything correct.*



success : 329 words used.

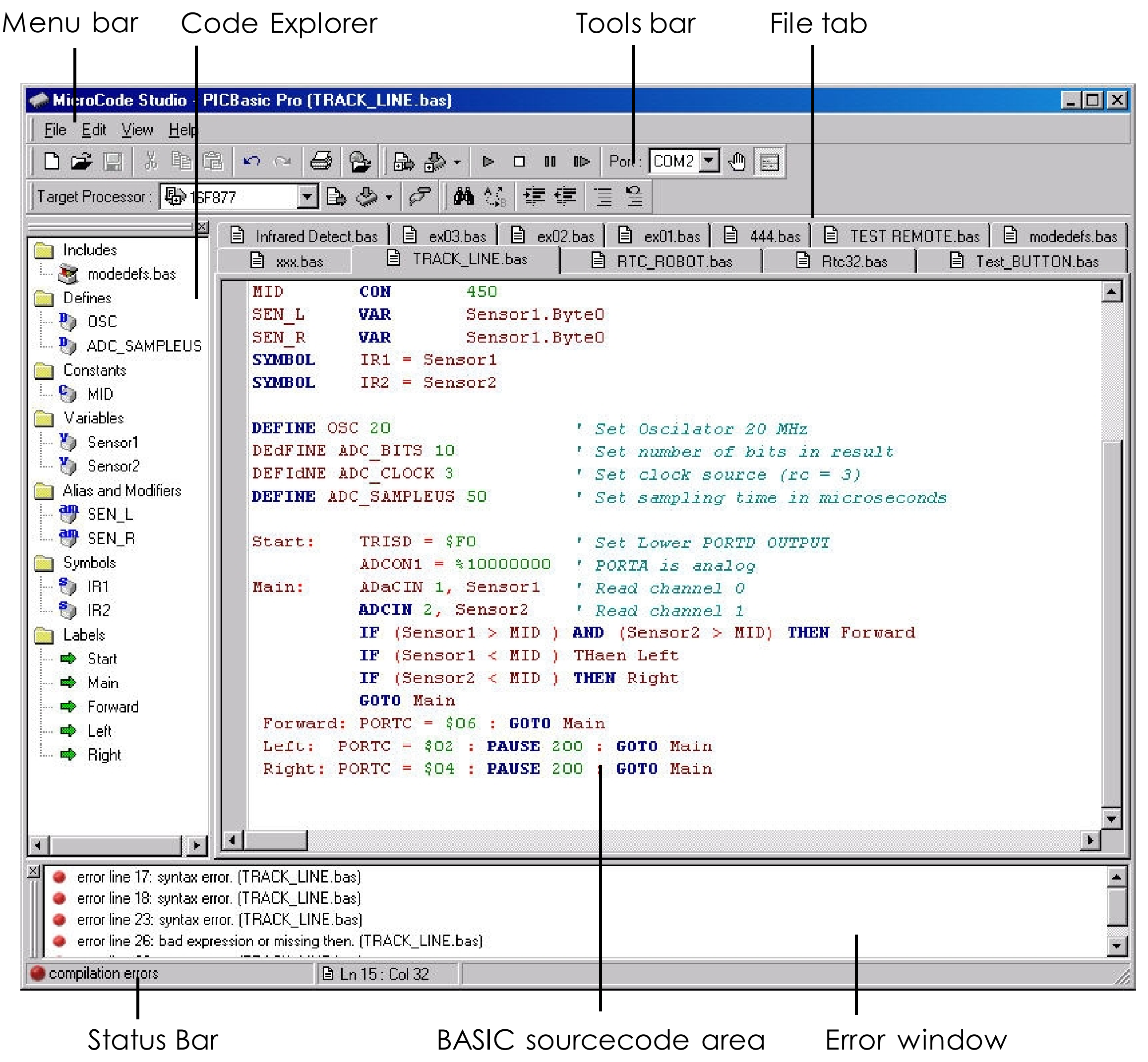
1.3.2.3 Microcode Studio element

The main element of Microcode Studio includes :

1. **Code area** - It provides color syntax and font's format to separate Instructions, Label and Comment. It helps user to understand easier.

2. **Code Explorer window** - It allows user to automatically jump to include files, defines, constants, variables, aliases and modifiers, symbols and labels that are contained within your source code.

3. **Error window** - After compile the sourcecode, may be the error will happen. This window will show all error. User can click at the error line to access for editing easier.



```
MID       CON       450
SEN_L     VAR       Sensor1.Byte0
SEN_R     VAR       Sensor1.Byte0
SYMBOL    IR1 = Sensor1
SYMBOL    IR2 = Sensor2

DEFINE OSC 20                   ' Set Oscilator 20 MHz
DEdFINE ADC_BITS 10             ' Set number of bits in result
DEFIdNE ADC_CLOCK 3             ' Set clock source (rc = 3)
DEFINE ADC_SAMPLEUS 50          ' Set sampling time in microseconds

Start:    TRISD = $F0           ' Set Lower PORTD OUTPUT
          ADCON1 = %10000000    ' PORTA is analog
Main:     ADaCIN 1, Sensor1     ' Read channel 0
          ADCIN 2, Sensor2      ' Read channel 1
          IF (Sensor1 > MID ) AND (Sensor2 > MID) THEN Forward
          IF (Sensor1 < MID ) THaen Left
          IF (Sensor2 < MID ) THEN Right
          GOTO Main
 Forward: PORTC = $06 : GOTO Main
 Left:  PORTC = $02 : PAUSE 200 : GOTO Main
 Right: PORTC = $04 : PAUSE 200 : GOTO Main
```

```
error line 17: syntax error. (TRACK_LINE.bas)
error line 18: syntax error. (TRACK_LINE.bas)
error line 23: syntax error. (TRACK_LINE.bas)
error line 26: bad expression or missing then. (TRACK_LINE.bas)
```

Figure 1-4 Microcode Studio element

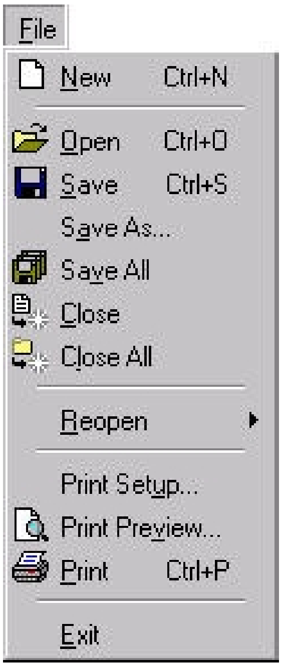
4. **Status bar** - It show the status by Red and Green LED symbols.

5. **Tools bar** - This bar will collect Edit and Compile instructions to help user to use program more comfortable.

6. **Serial Communications Window** - It is used for displaying the serial data that send out from microcontroller via DEBUG or SEROUT command. User can define all parameter.

1.3.2.4 Important command button of Microcode Studio

File



NEW (Ctrl+N) Creates a new document. A header is automatically generated, showing information such as author, copyright and date. To toggle this feature on or off, or edit the header properties, you should select editor options. When a new document is created, the default target processor device is set to the 16F877. To change the default, edit the file 'default.ini', located in your main installation folder. You need to restart MicroCode Studio after making the changes.

Open (Ctrl+O) Displays a open dialog box, enabling you to load a document into the MicroCode Studio IDE. If the document is already open, then the document is made the active editor page.

Save (Ctrl+S) Saves a BASIC code to disk.

Save As - Save a code and change the other filename to disk.

Save All - Save all the opened file to disk.

Close-Closes the currently active code

Close All -Closes all editor documents

Reopen - Displays a list of Most Recently Used (MRU) files.

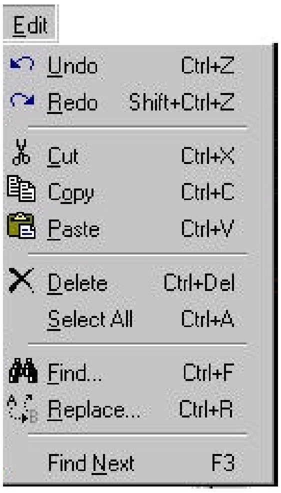
Print Setup - Displays a print setup dialog.

Print Preview - Displays a print preview window

Print - Prints the currently active editor page

Exit-Exit MicroCode Studio

Edit



Undo (Ctrl+Z) - Cancels any changes made to the currently active document page.

Redo (Shift+Ctrl+Z) -Reverse an undo command.

Cut (Ctrl+X) - Cuts any selected text from the active document page and places it into the clipboard.

Copy (Ctrl+C) Copies any selected text from the active document page and places it into the clipboard.

Paste (Ctrl+V) - Paste the contents of the clipboard into the active document page.

Delete (Ctrl+Del)- Deletes any selected text.

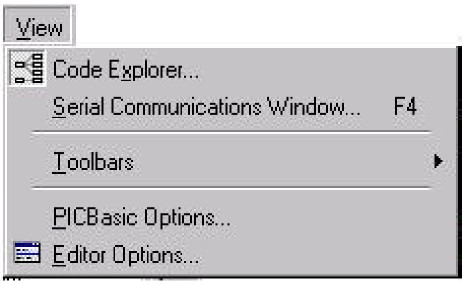
Select All (Ctrl+A)- Selects the entire text in the active document page.

Find (Ctrl+F) - Displays a find dialog.

Replace (Ctrl+R)- Displays a find and replace dialog.

Find Next (F3)- Automatically searches for the next occurrence of a word.

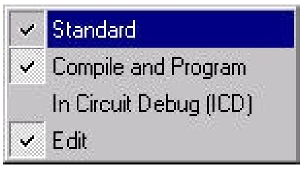
View



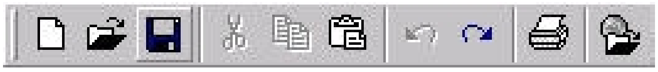
Code Explorer - Display or hide the code explorer window.

Serial Communications Window (F4) - Displays the Serial Communicator software. For more information, see the online help supplied with Serial Communicator.

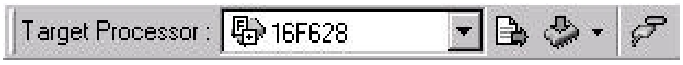
Toolbars - Display or hide the Standard, edit, compile and program and ICD toolbars.



Standard: Include standard tools such as New, Open or Save and etc.

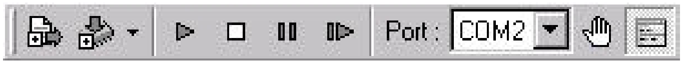


Compile and Program : Includes tools for compile, Select target microcontroller and open the Serial Communications



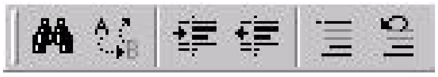
Target Processor : 16F628

In Circuit Debug (ICD) : Includes tools for In-Circuit Debugger. Require a hardware ICD (Do not support PICBASIC PRO compiler Demo version)

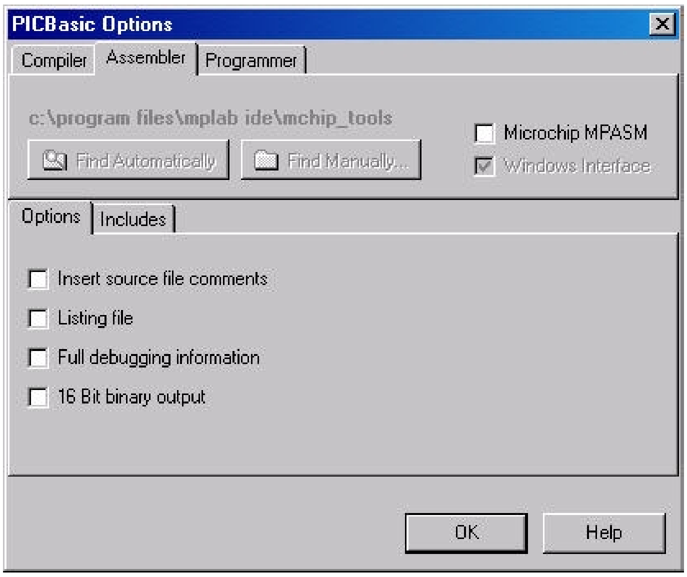


Port : COM2

Edit : Tools for editing source program. Includes Find, Replace, and Make the comment etc.



PICBASIC OPTION - Select and find folder that install PICBASIC PRO compiler and Programmer software.



PICBasic Options

c:\program files\mplab ide\mchip_tools

□ Microchip MPASM

☑ Windows Interface

□ Insert source file comments

□ Listing file

□ Full debugging information

□ 16 Bit binary output

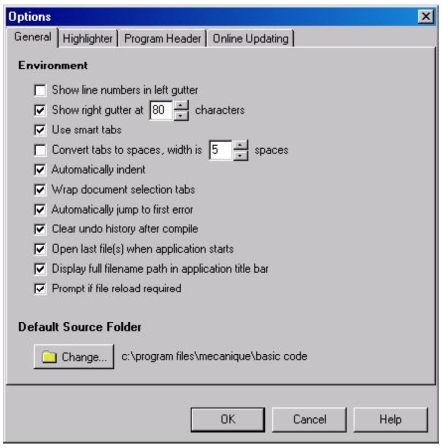
Compiler : Select the compiler folder

Assambler : Select the Assembler. Normally Microcode Studio select PM.EXE file for assembling the code. Then not nescessory to change this option.

Programmer : Select the Programmer software folder

Editor Option - Displays the main editor options dialog. Includes :

General : Set default parameter



Options

General | Highlighter | Program Header | Online Updating

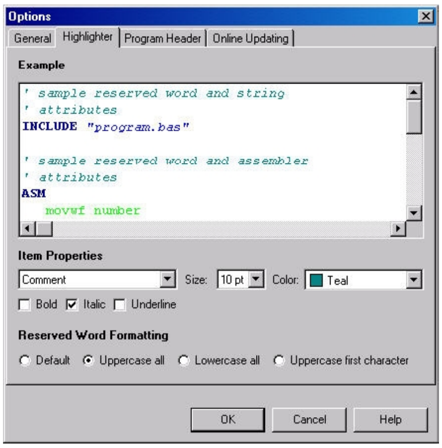
Environment

- □ Show line numbers in left gutter
- ☑ Show right gutter at 80 characters
- ☑ Use smart tabs
- □ Convert tabs to spaces, width is 5 spaces
- ☑ Automatically indent
- ☑ Wrap document selection tabs
- ☑ Automatically jump to first error
- ☑ Clear undo history after compile
- ☑ Open last file(s) when application starts
- ☑ Display full filename path in application title bar
- ☑ Prompt if file reload required

Default Source Folder

Change... c:\program files\mecanique\basic code

Highlighter : Adjust color and fonts for color syntax separate.



Options

General | Highlighter | Program Header | Online Updating

Example

```
' sample reserved word and string
' attributes
INCLUDE "program.bas"

' sample reserved word and assembler
' attributes
ASM
   movwf number
```

Item Properties

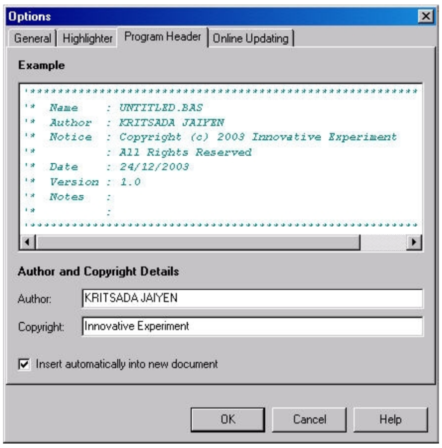
Comment Size: 10 pt Color: 🟦 Teal

□ Bold ☑ Italic □ Underline

Reserved Word Formatting

○ Default ● Uppercase all ○ Lowercase all ○ Uppercase first character

Program Header : Define the header comment for any sourcecode.



Options

General | Highlighter | Program Header | Online Updating

Example

```
'*******************************************************
'*  Name    : UNTITLED.BAS
'*  Author  : KRITSADA JAIYEN
'*  Notice  : Copyright (c) 2003 Innovative Experiment
'*          : All Rights Reserved
'*  Date    : 24/12/2003
'*  Version : 1.0
'*  Notes   :
'*          :
'*******************************************************
```

Author and Copyright Details

Author: KRITSADA JAIYEN

Copyright: Innovative Experiment

☑ Insert automatically into new document

Online Updating : Select for checking the update software from manufacturer's website.

1.3.3 Miracle PIC : Programmer software for PIC microcontroller

This is one of PIC microcontroller programmer software. Developed by Innovative Experiment Co.,Ltd. User can free download from www.inex.co.th. Miracle PIC needs a free computer's Parallel port and can interface with Microcode Studio. User can compile and program the code into the microcontroller in one click.

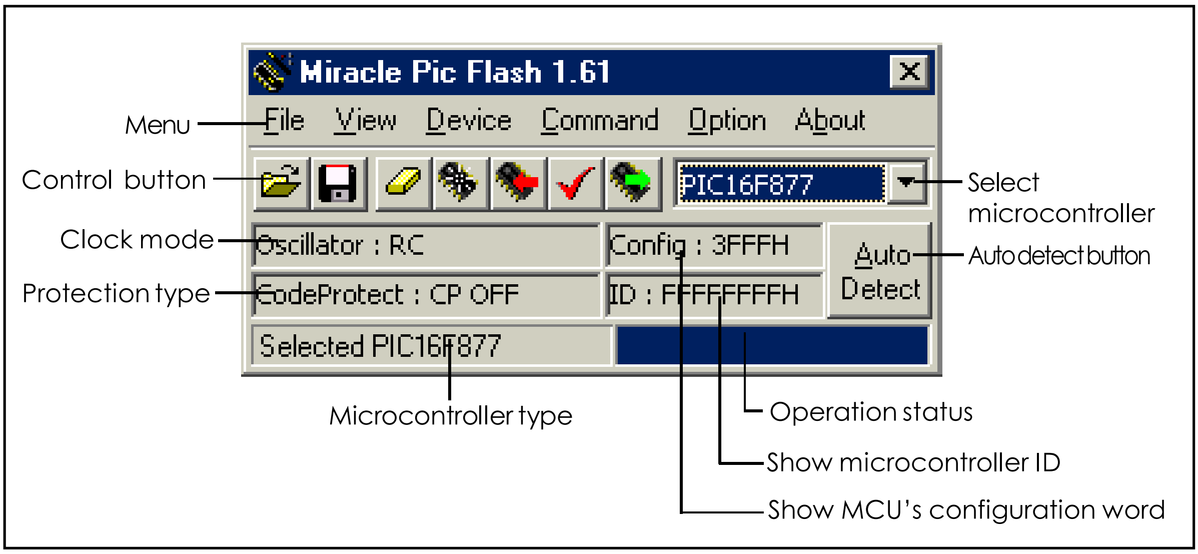
1.3.3.1 Installation

Enter to folder Tool à Miracle_PIC in Robo-PICA bundled CD-ROM. Double-click at mPICSetup161.exe (numver of version could be change depends on the latest version). The installation window will appear. Click OK button to begin the installation and wait until installation complete.

User can run the Miracle PIC by select Start à Program → Miracle PIC → Miracle PIC Flash. The main window will appear following the figure 1-5

1.3.3.2 Main Control Buttom

Open	: Open HEX file
Save	: Save source code in HEX file
PROGRAM	: Program all data into program memory and EEPROM
VERIFY	: Test program; compare data in buffer with MCU's data
READ	: Read data from selected microcontroller and also read parameter



PIC16F877

Oscillator : RC

Config : 3FFFH

CodeProtect : CP OFF

ID : FFFFFFFFH

Selected PIC16F877

Figure 1-5 : Miracle-PIC Flash software main window

BLANK CHECK : Check blank data inside microcontroller

ERASE : Delete data in program memory of selected microcontroller. This function used with flash microcontroller memory or EEPROM only.

PIC16F877

Select microcontroller : Just click at the arrow. User will see many listing.

1.3.3.3 Menu description

File

Open file - To open file.HEX

Save file As - To save data in another file name

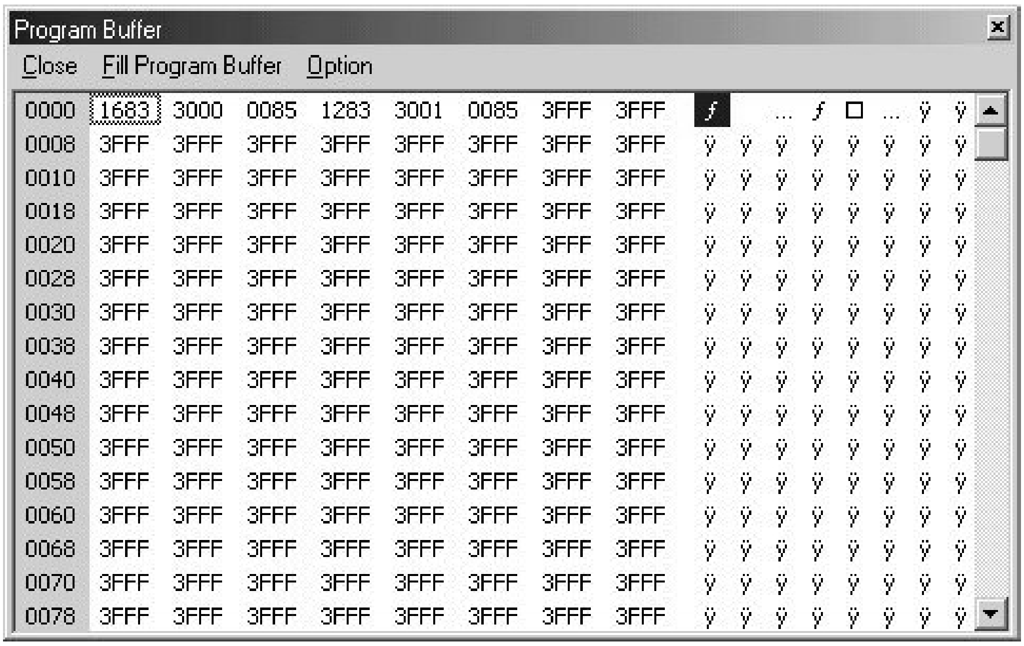
Recent files - To open recently file

Exit - Quit

View

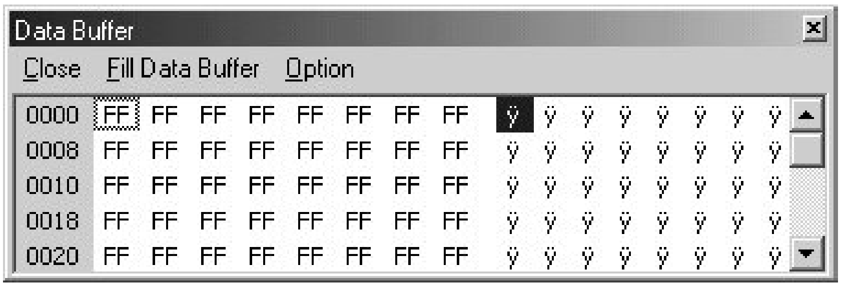
To see all parameter of PIC microcontroller

Program Buffer (F11) - shows code in program memory (figure 1-6). User can edit by select memory position and input value directly or use Fill Progran Bufer Function



```
0000 1683 3000 0085 1283 3001 0085 3FFF 3FFF  ƒ   … ƒ □ … ÿ ÿ
0008 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0010 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0018 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0020 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0028 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0030 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0038 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0040 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0048 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0050 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0058 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0060 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0068 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0070 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0078 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
```

Figure 1-6 : Program buffer window, user can see all code and edit any address



Data Buffer

```
0000 FF FF FF FF FF FF FF FF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0008 FF FF FF FF FF FF FF FF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0010 FF FF FF FF FF FF FF FF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0018 FF FF FF FF FF FF FF FF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
0020 FF FF FF FF FF FF FF FF  ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ
```

Figure 1-7 : Data buffer window, user can see all data in EEPROM within microcontroller and edit any address

Data Buffer (F12) - shows all data EEPROM (figure 1-7). User can edit data by select memory position and input value directly or use Fill Progran Buffer Function

Configuration (F9) - shows detail of PIC configuration. User can enable or disable all parameter.

Count Program - show number of programmed microcontroller *(not use this function in RBX-877 robot experiment board)*

Stay on top - Make the program always on the top

Configuration Setting

Before programing PIC microcontroller, always fix configuraruon, go to View à Configuration. This is detail of them.

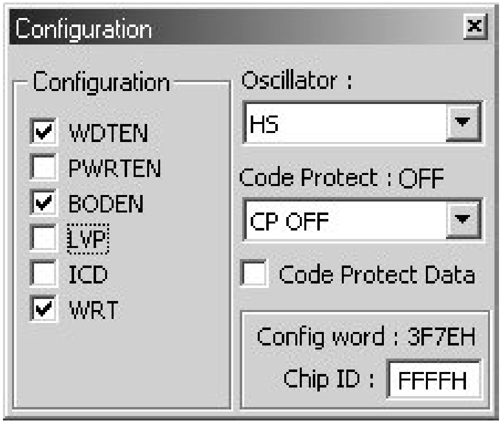
Oscillator - select clock mode, there are 4 types (for PIC16F877-20/P)

LP - low cystal (not use in the RBX-877 board)

XT - normal cystal not over 4 MHz *(not use in the RBX-877)*

HS - Over 4 MHz high frequency xtal ***(select for RBX-877)***

RC - External RC circuit (not use in RBX-877 board)



Configuration

Configuration
☑ WDTEN
☐ PWRTEN
☑ BODEN
☐ LVP
☐ ICD
☑ WRT

Oscillator : HS

Code Protect : OFF

CP OFF

☐ Code Protect Data

Config word : 3F7EH

Chip ID : FFFFH

Figure 4 : Configuration window

Code protection - Use for selceting protection mode. It can protect data in PIC16F877 microcontroller is 4 modes include : ALL, 1F00H-1FFFH, 1000H-1FFFH and OFF. Select CPD box for protection data in EEPROM.

Configuration - To select special ability of PIC miocrocontroller, it depends on type of microcontroller. For PIC16F877 has 6 types;

WPT (watchdog Timer)

PWRT (Power-up Timer)

BODEN (Brown-out Reset)

LVP (Low Voltage program) - Enable programming to PIC microcontroller with low voltage+5V instead of +12V (Should not use with JX-877)

ICD (In-circuit Debugger) - Edit program memory in position OX0100(100H) up via RB6 and RB7 so RB6 and RB7 is not work(no use in JX-877)

WRT (Flash progran write) - If select this function, users can program data into flash program memory via resistor EECON. On the other hand, no select this, the program in the memory have to do via RB6 and RB7 in normally.

Device

Select type of PIC microcontroller

Command

Arrange reading, writing and testing data in microcontroller memory

Program Chip (F4) - Program code and data to microcontroller as PROGRAM buttom

Verify Chip (F5) - Check writing data in PIC microcontroller correct or incorrect as VERIFY buttom

Read Chip (F6) - Read data from microcontroller as READ buttom

Erase Chip (F7) - Delete data in PIC microconrtoller as ERASE buttom

Blank Chip (F8) - Check blank data in PIC microcontroller memory as BLANK CHECK button

Option

Separate 4 parts;

Part1

Program/Verify Program - check latest programming in microcontroller with buffer.

Program/Verify Data - check data in microcontroller memory after latest programming

Program/Verify ID - checking ID value in microcontroller after latest programming *(not use in JX-877)*

Program/Verify Configuration - check configuration value after latest programming

Part2 Special feature in Miracle-PIC programmer software

Verfy while program - check data while program microcontroller. If occur the faults, program will stop in suddenly.

Auto verify after program - check data after completely programming automatic.

Auto erase before program - delete data before programming

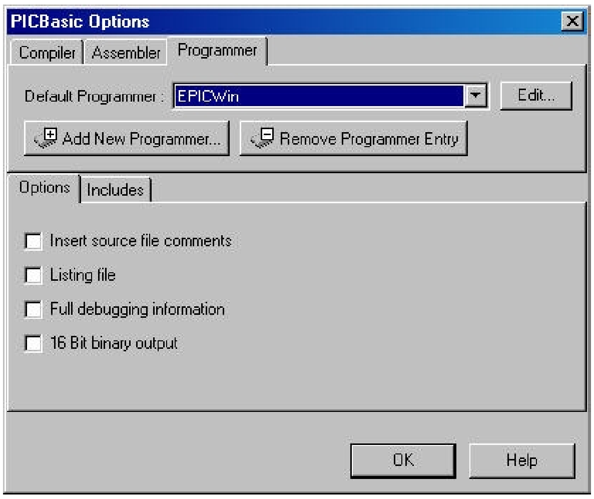
Program all address - program data in every address

Part3 **Configuration erase message** - selcet warning window

Part4 **Parallel port** - select parallel port LPT1 to LPT3

1.3.3.4 How to interface Miracle PIC with Microcode Studio

Step 1 : Open Microcode Studio. Select menu bar VIEW → PICBasic Option then select the Programmer tab

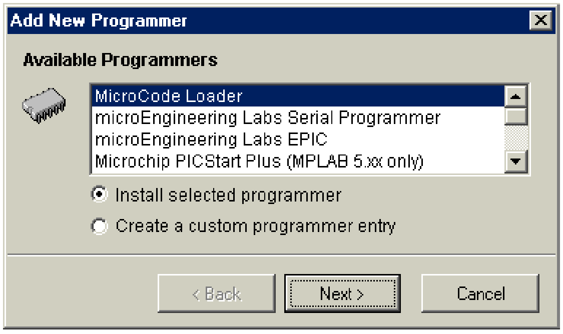


PICBasic Options

Default Programmer : EPICWin

□ Insert source file comments
□ Listing file
□ Full debugging information
□ 16 Bit binary output

Step 2 : Click the Add New Programmer... button to add the new programmer. The Add Ndew Programmer window will appear

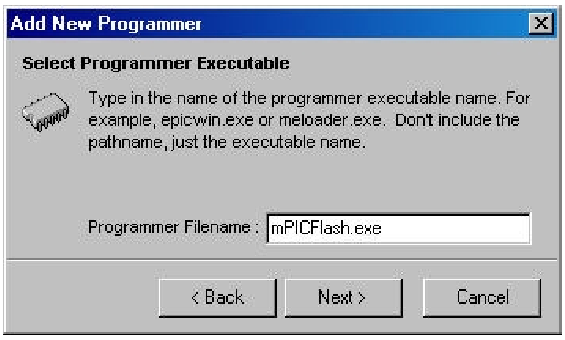


MicroCode Loader
microEngineering Labs Serial Programmer
microEngineering Labs EPIC
Microchip PICStart Plus (MPLAB 5.xx only)

● Install selected programmer
○ Create a custom programmer entry

Step 3 : Select Create a custom programmer entry then click Next > button. It will show a window for defining the programmer's name. Put Miracle PIC name and click Next > button to next step.

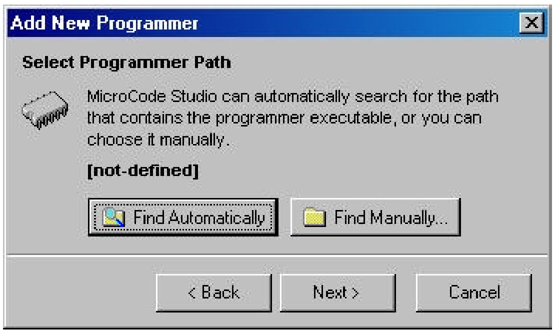
Step 4 : Select the programmer software execute file. For Miracle PIC is *mPICFlash*.exe. Type this filename into the message box. After that click Next > button to next step.



Type in the name of the programmer executable name. For example, epicwin.exe or meloader.exe. Don't include the pathname, just the executable name.

Programmer Filename : mPICFlash.exe

Step 5 : Find the folder that install Miracle PIC software. User can find automatically by clicked Find Automatically button or manually by clicked Find Manually... button

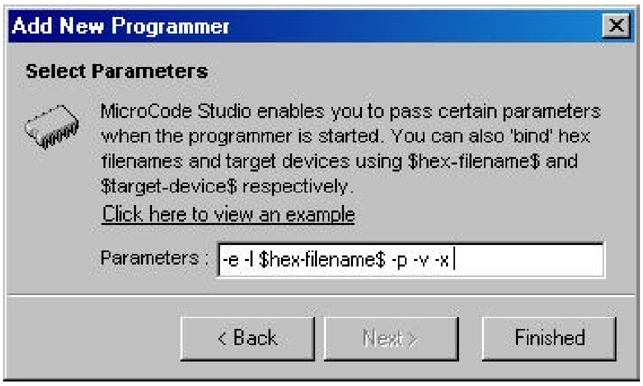


MicroCode Studio can automatically search for the path that contains the programmer executable, or you can choose it manually.

[not-defined]

Step 6 : Select the parameters for passing important parameter from HEX filename to Miracle PIC in automatic. This step set to support one-click operation from compile to program the cose into microcontroller. The parameter is :

```
-e -l $hex-filename$ -p -v -x
```



Add New Programmer

Select Parameters

MicroCode Studio enables you to pass certain parameters when the programmer is started. You can also 'bind' hex filenames and target devices using $hex-filename$ and $target-device$ respectively.

Click here to view an example

Parameters : -e -l $hex-filename$ -p -v -x

Parameter	Description
-e	Erase data in the program memory before program.
-l $hex-filename$	Load the target .hex file from Microcode Studio
-p	Program the chip
-v	Verify the chip after programming
-x	Exit from Miracle PIC after finish

Click Finished button for finishing this setting.

Chapter 2

RBX-877

PIC16F877 Robot Experiment Board

Robo-PICA robot kit is controlled b RBX-877 (PIC16 877 Robot Experiment board). The main microcontroller is PIC16 877. The figure 2-1 shows operating diagram of RBX-877 board. In this chapter will present the operation of RBX-877 board and some example experiment. Builders must read and test following all experiments to reference for building and programming the robot in next chapter.

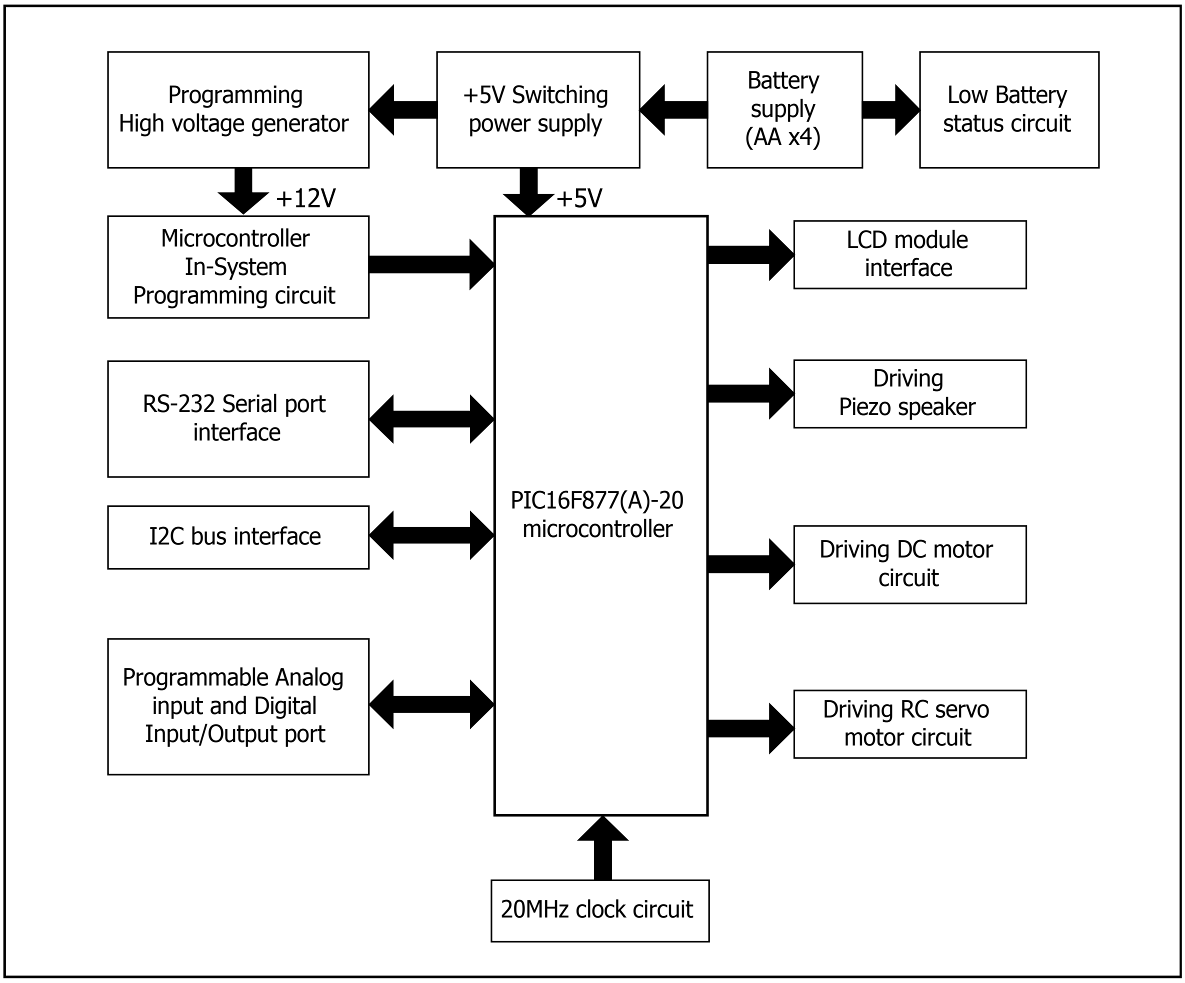

Programming High voltage generator
+5V Switching power supply
Battery supply (AA x4)
Low Battery status circuit
+12V
+5V
Microcontroller In-System Programming circuit
RS-232 Serial port interface
I2C bus interface
Programmable Analog input and Digital Input/Output port
PIC16F877(A)-20 microcontroller
LCD module interface
Driving Piezo speaker
Driving DC motor circuit
Driving RC servo motor circuit
20MHz clock circuit

Figure 2-1 RBX-877 Robot expriment board's block diagram

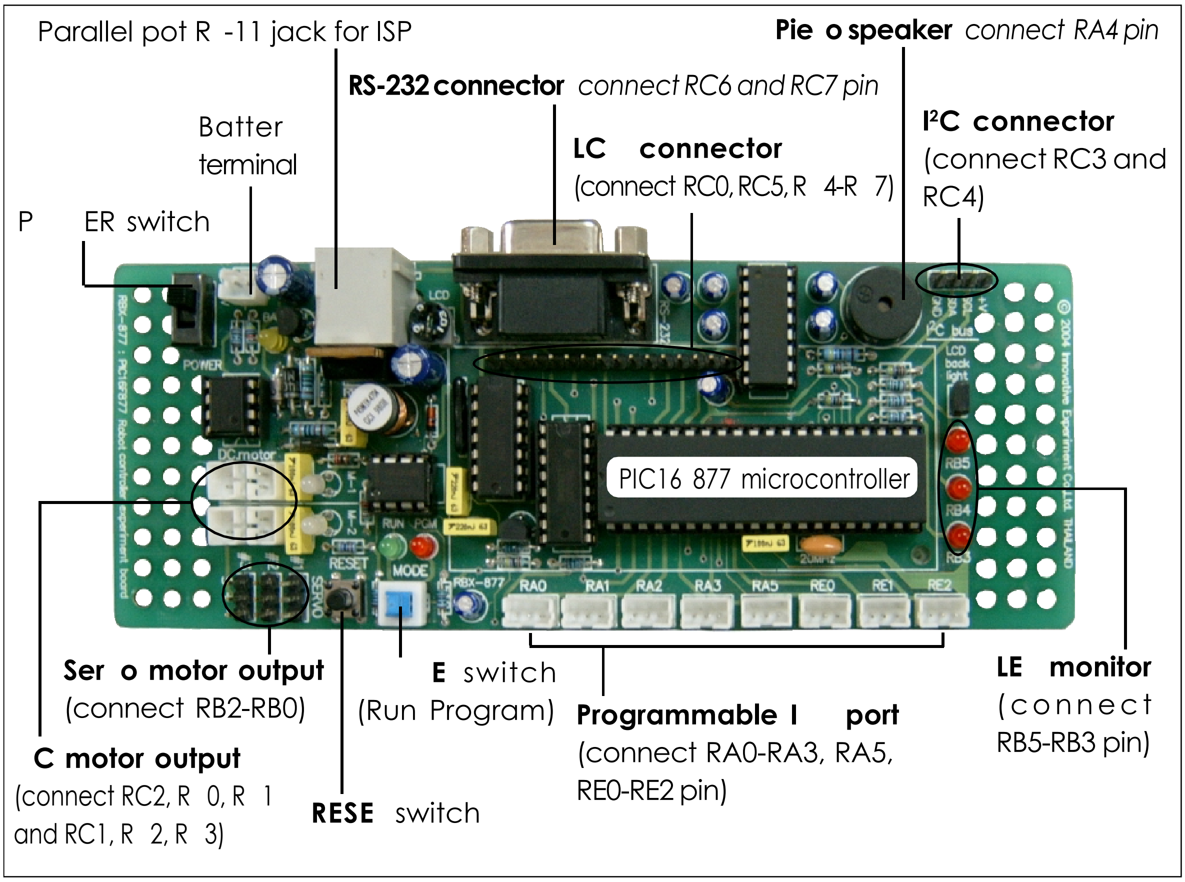

Parallel pot R -11 jack for ISP
Pie o speaker *connect RA4 pin*
RS-232 connector *connect RC6 and RC7 pin*
Batter terminal
LC connector (connect RC0, RC5, R 4-R 7)
I²C connector (connect RC3 and RC4)
P ER switch
PIC16 877 microcontroller
Ser o motor output (connect RB2-RB0)
E switch (Run Program)
Programmable I port (connect RA0-RA3, RA5, RE0-RE2 pin)
LE monitor (connect RB5-RB3 pin)
C motor output (connect RC2, R 0, R 1 and RC1, R 2, R 3)
RESE switch

Figure 2-2 RBX-877 board layout

2.1 echnical features

- Controlled b PIC16 877(A) 20 Hz icrocontroller with 8 word memor
- ownload program via Parallel port, selected b ode switch
- Serial port interface
- LC 16x2 displa
- Piezo speaker
- 3-LE monitor
- rive 2- C motors 4.5 to 6 and 3-RC Servo motors (in range 4.8 to 6)
- 8-Programmable port support all analog inout and digtial input output
- Suppl voltage from 4 of AA batteries (Rechargable 1700mAH recommended)
- 2.375 x 6.25 Inches size

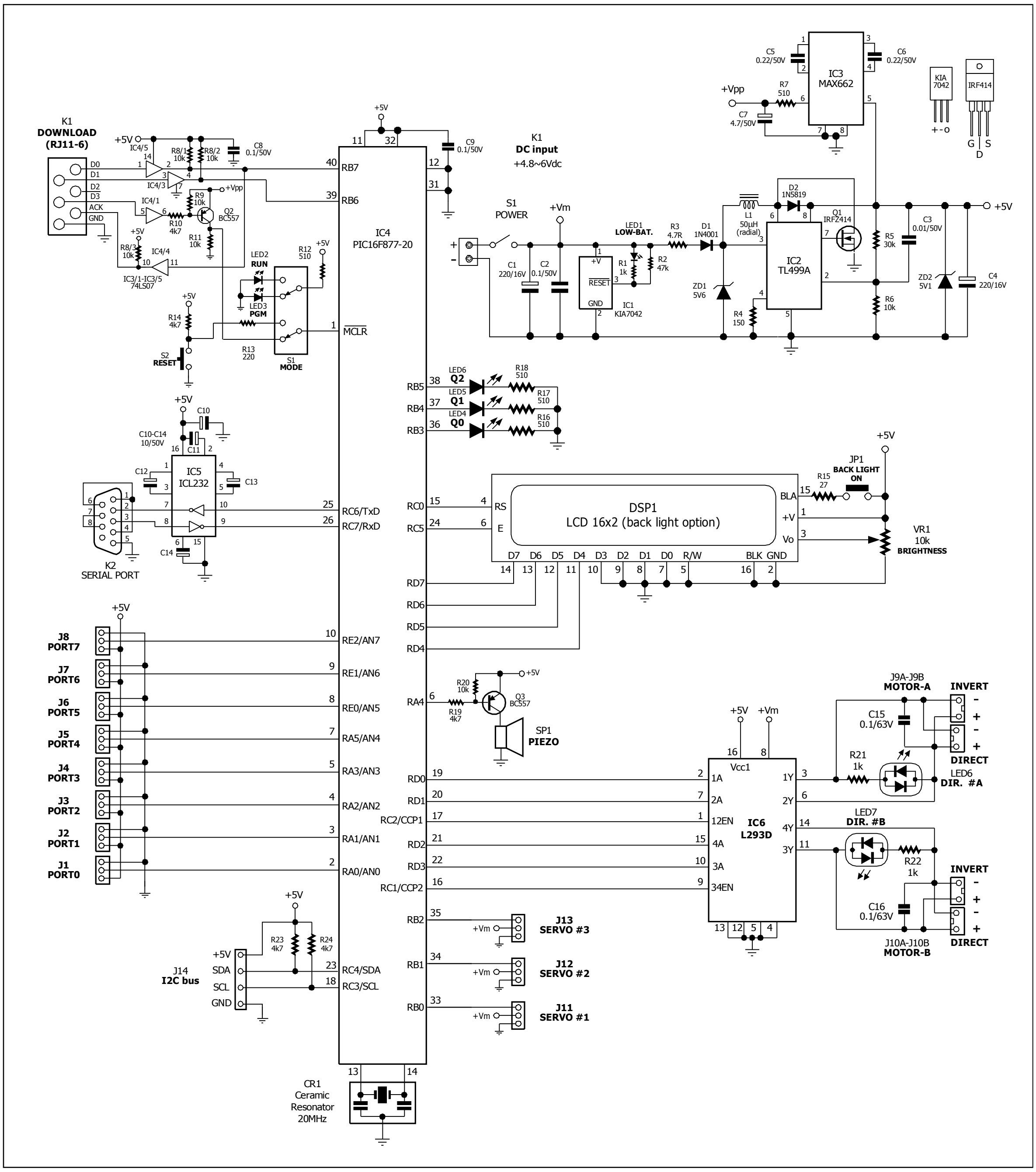


Figure 2-3 Schematic diagram of RBX-877 Robot Experiment board

2.2 RBX-877 board circuit description

2.2.1 icrocontroller circuit

The heart of this board is PIC16 877 microcontroller. The 20 Hz ceramic resonator, CR1 is used to make the 20 Hz clock for PIC16 877

2.2.2 Power suppl

The RBX-877 board contains a step-up swithching power suppl to suppl 5 regulated for PIC16 877. Although the level of batter will decrease in driving motor. This switching power suppl circuit will maintain the 5 for microcontroller until batter voltage level donw to 1.5

S1 is on-off switch to suppl the voltage from batteries to RBX-877 board. IC1- IA7042 is used to detect the suppl voltage. If lower 4.2 , output pin of IC1 will be logic 0 . It cause LE 1 turn-on for report L -BAT status. R3, 1 and Z 1 ard used to limit the input voltage to IC2 not over 5.1

IC2 is a switching powersuppl IC, TL4 A. It can support input voltage 4.2-5.6 range for regulating 5 suppl voltage. Z 2 is used to limit output voltage of TL4 not over 5 .

2.2.3 In-S stem Programming circuit

5 suppl is send to AX662 for switching up to 12.5 high voltage for programming the PIC16 877 microcontroller

S2, E switch is used to select operaion mode of RBX-877 board Run or Program. In program mode selected, the red LE will be turn-on and CLR pin of PIC16 877 is connected to 12.5 from AX662. RB6 and RB7 pin of PIC16 877 are connected to clock and data signal of in-s stem programming. In rn mode selected, the green LE will be turn-on and CLR pin is connected to RESET switch fot resetting the microcontroller s operation.

2.2.4 ispla circuit

Character display : The RBX-877 board provides LC module connector. It supports 16 characters 2-lines LC . PIC16 877 s R 4-R 7 pin is assigned to 4- 7 data pins, RC5 to E pin and RC0 to RS pin for selection data mode. R1 is used to contrast adjustment of LC screen. In case using Back-light LC , it provides a jumper to control the LE back-light of LC

LED monitor RBX-877 board has 3 fo general purpose LE . The are connected to RB3, RB4 and RB5 of PIC16 877 microcontroller via the current limiied resistor.

Sound output RBX-877 has a sound driver circuit. Connect RA4 pin to driver circuit and piezo speaker. This circuit can drive audio fre uenc signal. However the piezo speaker has the resonance fre uenc in range 1kHz to 3kHz.

2.2.5 Programmable port

The RBX-877 board provides 8-programmable port for mutlti-purpose. It includes RA0-RA3, RA5, RE0-RE2 pin. All port pin cab program to 3 functions as

1. Analog input - to get analog signal to A converter circuit inside microcontroller. Input voltage range is 0 to 5 . Converter resolution is 10-bit.

2. igital input - to get digital signal from digtial de ice and switch

3. igital output - to dri e digital signal logic 0 and 1 to external device.

In default all port will be set to analog input port.

n RBX-877 board provides all ports in 3-pin PCB connector. Each connector includes 5 and .

2.2.6 RS-232 serial port interface for serial data communication

Buiders can make the serial data communication from RBX-877 board to computer s RS-232 serial port. PIC16 877 microcontroller provides RC6 and RC7 pin ART module port pin for this purpose. Serial signal from PIC16 877 s ART must convert to RS-232 level b IC5, AX232 and feed to 2 B- female connector.

2.2.7 C motor dri er circuit

The RBX-877 board use IC6, L2 3 H-bridge motor driver IC for driving C motor 2 channels. The suitable motor is 4.5-6 100-200mA.

otor A speed is conrtolled b R 0 with R 1 pin and enable b RC2 port. otor B speed is conrolled b R 2 with R 3 pin and enable b RC1. LE 6 and LE 7 are bi-color LE . The are used for showing the motor output status.

oltage is supplied to L2 3 includes 5 suppl voltage and otor suppl voltage (m). The m is conencted direct from batteries for powerful driving.

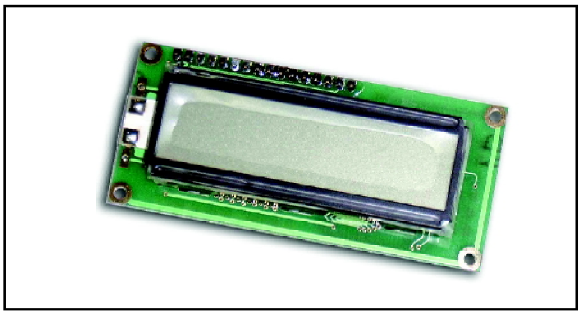


Figure 2-4 Character LCDmodule for RBX-877 board

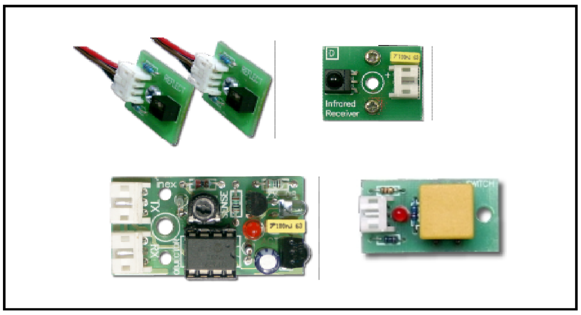


Figure 2-5 Sample of sensor and detector board that interface with RBX-877 board

2.2.8 RC ser o motor dri er circuit

The RBX-877 provides 3 port pin to driving RC servo motor. It includes RB0, RB1 and RB2 . RC servo motor suppl   get from the s stem batter . This driver cannot support high-current and high power RC servo motor. The suitable RC servo motor is 4.8 to 6   motor and need current consumption about 100-200mA.

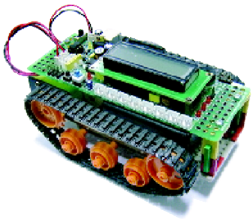
2.2.   I^{2}C connector

A wa   to expansion of RBX-877 board is using I2C bus connector. Because PIC16 877 has I2C bus module.   an   external device need I2C bus protocal such as Real-time clock, memor , A   and   A converter, port expansion device and etc. RC3 SCL and RC4 S   A of PIC16 877 are connected to I^{2}C bus connector includes   5   suppl   and   . R23 and R24 pull-up resistor are connected to theses port for setting idle state.

All example sourcecode select PIC16 877 microcontroller. See the code in line

```
@ DEVICE   PIC16F877,HS_OSC ' Use PIC16F877 and HS Oscilator
```

If change to PIC16F877A or another, must edit this code at this line *to make s re the correct microcontroller n mber and config ration*



Acti it 1

rite programs for testing RBX-877 Robot Experiment board

Procedure

This is standard step for testing the operation of RBX-877 board and robot programming.

1. pen icrocode Studio that interface PICBASIC PR compiler (demo or full version) and iracle PIC software read .

2. rite the experiment program following each sub-activit

3. Put 4 of AA batteries into Batter holder of RBX-877.

4. Connect the CX-6 cable from RBX-877 board to computer s parallel port

5. Connect the addition device following each activit to intereface connector on RBX-877 board after that turn-on P ER switch.

6. Select E switch to Program mode.(The red LE at P . on)

7. Compile the code and download HEX file to RBX-877 board.

8. Change the E switch to R mode for running the program within RBX-877 board.

Acti it 1-1 LE testing

See the figure A1-1, RB3,RB4 and RB5 of PIC16 877 connect to LE via current limited resistor 510 . or turning-on these LE s must send logic 1 to these port. and send logic 0 for turning-off.

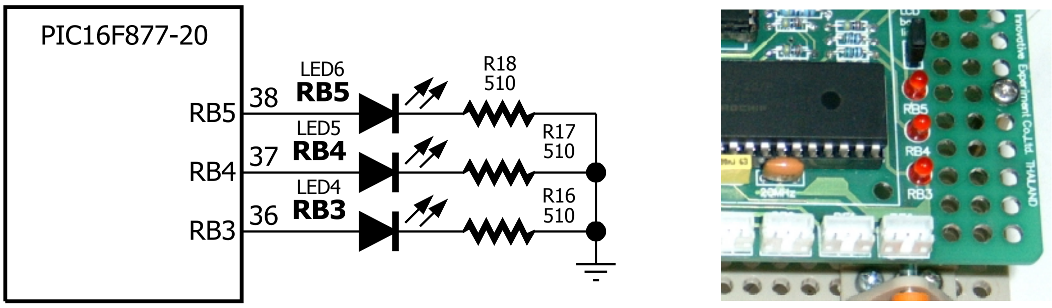


Figure A1-1 LED connection on RBX-877 board

```
' File : a0101.bas
' Description : Drive LED at RBb by HIGH command
@ DEVICE  PIC16F877,HS_OSC          ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                       ' Use Oscillator 20 MHz

HIGH PORTB.3                        ' Set LED at RB3 --> ON
LOOP: GOTO LOOP                     ' Still Here
```

Listing A1-1 Turn-on LED at RB3

A1.1.1 rite program following the listing A1-1 then compile and download to RBX-877 board. See the operation.

E at RB3 on

A1.1.2 Refer the listing A1-1, for more clearing of command operation. In the listing A1-2 shows the 2 commands to drive LE TP T P RTB.4 comamnd for assign to RB4 as output and set logic 1 to RB4 pin in the next line. See the listing A1-2

```
' File : a0102.bas
' Description : Drive LED at RB4 by basic method
@ DEVICE  PIC16F877,HS_OSC          ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                       ' Use Oscillator 20 MHz

OUTPUT PORTB.4                      ' RB4 --> OUTPUT
PORTB.4 = 1                         ' Set LED at RB4 --> ON
LOOP:      GOTO LOOP                ' Still Here
```

Listing A1-2 Control LED at RB4 port by simple technique. Set the port to output and send logic “1” to that port.

A1.1.3 Change the output data b b te data. See the listing 1-3. It assign the b te data to port B for select direction b **RISB** command

To set output direction write 0

To set input direction write 1

or assign the data, can write the data direct to port B

A1.1.4 rite the listing A1-3. Compile and download the code to RBX-877 board. bserve the operation.

E at RB3 RB4 and RB5 on and off similar 3-channels r nning light

```
' File : a0103.bas
' Description : Basic 3-dots running light
@ DEVICE  PIC16F877,HS_OSC         ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                      ' Use Oscillator 20 MHz
TRISB = %11000111                  ' RB3,RB4,RB5 --> OUTPUT
LOOP:PORTB = %00001000             ' LED at RB3 --> ON
           PAUSE 500               ' Delay Time 0.5 Sec
           PORTB = %00010000       ' LED at RB4 --> ON
           PAUSE 500               ' Delay Time 0.5 Sec
           PORTB = %00100000       ' LED at RB5 --> ON
           PAUSE 500               ' Delay Time 0.5 Sec
           GOTO LOOP
```

Listing A1-3 Make running light by byte data

Acti it 1-2 Reading digital data ia switch

See the figue A1-2, it shows schematic of the switch input board ZX-01. If switch not pressed, ATA point as logic 1 from pull-up resistor 10k . If switch pressed, ATA point will connect to ground. It cause ATA point is logic 0 . LE lights following switch pressed.

Reading switch input programming

The easiest wa to checking thw switch pressed in BASIC program of PICBASIC PR compiler is looping and check with I ...THE command. If switch pressed, the program will jump following the condition.

In writing the program, must select the port that interface the switch before. After that assign that port to digital input b wirte the data to setting into A C 1 register.

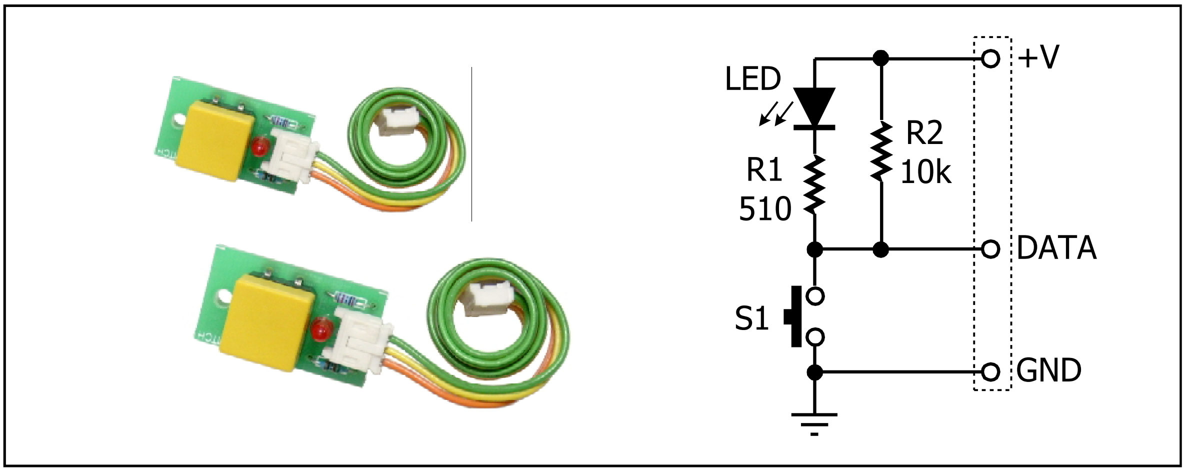


Figure A1-2 Switch input board schematic

esting

A1.2.1 Connect the switch input board ZX-01 to RA0 connector on RBX-877 board following the figure A1-3

A1.2.2 rite the listing A1-4 Compile and download the code to RBX-877 board.

A1.2.3 Press the switch and observe the operation.

isten so nd from the piezo speaker following the switch pressing

In this acti it use port A normall as analog port. hen access to A C 1 register for set the operation to digital input instead. he written alue is 7

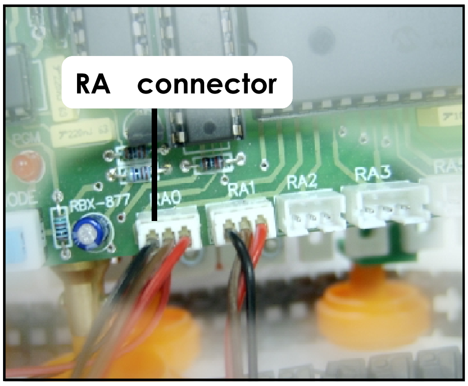


Figure A1-3 Shows the position of RA0 connector on RBX-877 board that connect the ZX-01 switch input board for reading digital value.

```
' File : a0104.bas
' Description : Reading digital input via switch board
@ DEVICE  PIC16F877,HS_OSC                  ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                               ' Use Oscillator 20 MHz
ÄDCON1 = $07                                ' Set PORTA & PORTE --> Digital I/O
LOOP:IF PORTA.0 = 0 THEN                    ' Test Switch at RA0 Press ?
           FREQOUT PORTA.4,300,1000         ' Show Sound on Piezo  0.3 Sec
     ENDIF
     GOTO LOOP                              ' Test Again
```

Listing A1-4 The BASIC program of reading digital value from the switch input board to control driving sound to piezo speaker. The operation is similar the door chime.

Acti it 1-3 Show message on LC module

The RBX-877 board provides the connector to interface LC moudule. The schematic diagram shows in the figure A1-4. ser must use this information to define in the BASIC program for PICBASIC PR compiler knows the port pin that use in this interface.

Interfacing use 6 port pins include R 4-R 7 for data pin 4- 7 in mode 4-bit interface, RC0 to RS pin and RC5 to E pulse pin. R pin of LC is connected to ground for onl writing all data to LC onl .

LC port definition of RBX-877 board for PICBASIC PR compiler

```
DEFINE LCD_DREG   PORTD  ' Set LCD Data port
DEFINE LCD_DBIT   4      '  Set starting Data bit (0 or 4) if 4-bit bus
DEFINE LCD_RSREG  PORTC  ' Set LCD Register Select port
DEFINE LCD_RSBIT  0      ' Set LCD Register Select bit
DEFINE LCD_EREG   PORTC  ' Set LCD Enable port
DEFINE LCD_EBIT   5      ' Set LCD Enable bit
DEFINE LCD_BITS   4      ' Set LCD bus size (4 or 8 bits)
DEFINE LCD_LINES  2      ' Set number of lines on LCD
DEFINE LCD_COMMANDUS 2000 ' Set command delay time in microsecond
DEFINE LCD_DATAUS 50     ' Set data delay time in microsecond
```

EFI E LC command must put in the header of program alwa before use LC command.

Command is sed to send message to C as **LC**

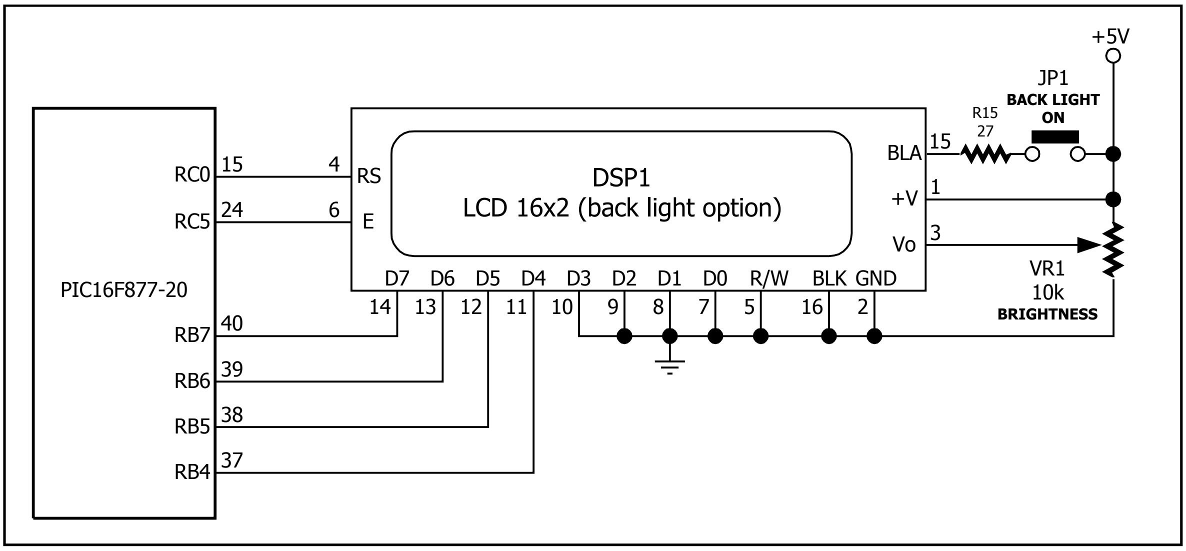


Figure A1-4 LCD interface schematic of RBX-877 board

To control LC must send command and data

In writing command Begins with write **FE** code to inform compiler to the next data as command. or example, Clear screen command is E and data for clearing screen command is 01. Show this example program in the listing A1-5.

esting

A1.3.1 Connect LC module onto 16-pin LC connector on RBX-877 board. If need to use the back -light, put jumper at LC backlight position.

A1.3.2 rite the listing A1-5 Compile and download the code to RBX-877 board. bserve the operation.

At C mod le show message Innovative *on the pper line and* Experiment *on the lower line*

```
' File : a0105.bas
' Description : Show message on LCD module
@ DEVICE  PIC16F877,HS_OSC          ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                       ' Use Oscillator 20 MHz

DEFINE     LCD_DREG    PORTD        ' Set LCD Data port
DEFINE     LCD_DBIT    4            ' Set starting Data bit (0 or 4) if 4-bit bus
DEFINE     LCD_RSREG   PORTC        ' Set LCD Register Select port
DEFINE     LCD_RSBIT   0            ' Set LCD Register Select bit
DEFINE     LCD_EREG    PORTC        ' Set LCD Enable port
DEFINE     LCD_EBIT    5            ' Set LCD Enable bit
DEFINE     LCD_BITS    4            ' Set LCD bus size (4 or 8 bits)
DEFINE     LCD_LINES   2            ' Set number of lines on LCD
DEFINE     LCD_COMMANDUS 2000       ' Set command delay time in us
DEFINE     LCD_DATAUS  50           ' Set data delay time in us
Main:      LCDOUT $FE,$01           ' Send Command To Clear Screen
           LCDOUT "Innovative"      ' Show Text on Line 1
           LCDOUT $FE,$C0           ' Set Cursor on Line 2
           LCDOUT "Experiment"      ' Show Text on Line 2
END
```

Listing A1-5 Program for displaying message on LCD module of RBX-877 board

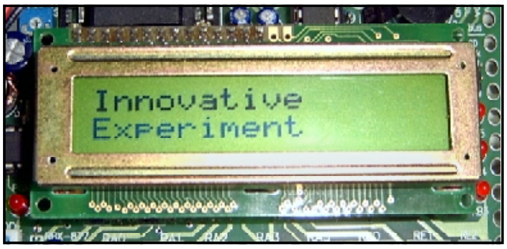


Figure A1-5 Show the operationn of the listing A1-5 at LCD module on RBX-877 board

If cannot see the character on C screen may be need to adj st the brightness at variable resistor in Brightness position on RBX-877 board by a small Philips screw driver

Acti it 1-4 he serial data communication with computer

The schematic of RS-232 serial port interfacing of RBX-877 board is shown in the figure A1-6. However in computer serial port interface, need some software tool as

A per terminal

H per Terminal is the populated terminal program. ormall install with indows S together. ser can run b select menu Accessories Communications H per Terminal. Its icon is HyperTerminal . H per Terminal has the setting step see the figure A1-7

(B) Serial communication Window in Microcode Studio

In the icrocode Studio, it provides the terminal window fro serail port communication. It s called Serial communication indow. ser can run b enter to menu iew Serial Communication indow in icrocode Studio window or press 4 button or click at icon on the main window of icrocode Studio. The figure A1-7 shows the Serial communication window of icrocode Studio. ser can change and set an value directl .

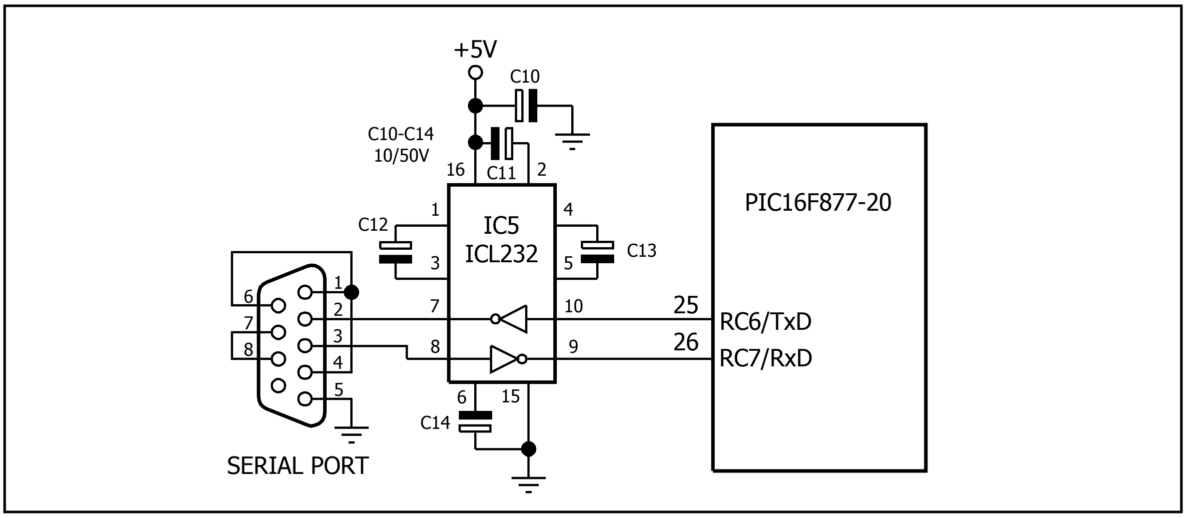
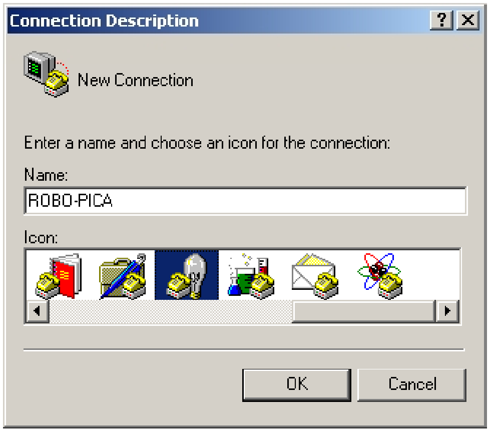


Figure A1-6 RS-232 Serial port interface schematic of RBX-877 board



Connection Description

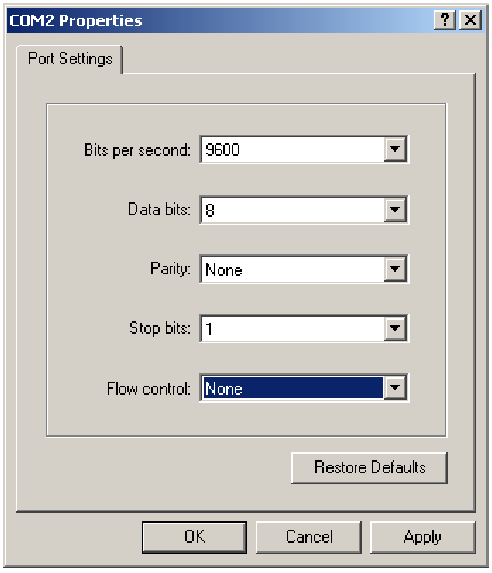
New Connection

Enter a name and choose an icon for the connection:

Name: ROBO-PICA

Icon:

Step 1 irst time to run, t pe the connection name in to ame box then press button



COM2 Properties

Port Settings

Bits per second: 9600

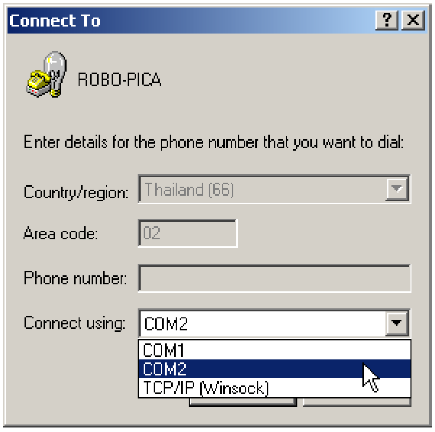
Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Step 3 Set the baudrate and data format to e ual at microcontroller setting. ormall is 600 8 1 (Baudrate 600, data is 8-bit, no parit and stop is 1 bit)



Connect To

ROBO-PICA

Enter details for the phone number that you want to dial:

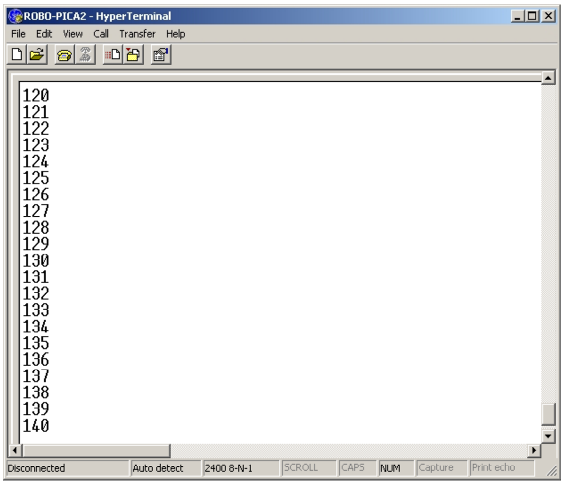
Country/region: Thailand (66)

Area code: 02

Phone number:

Connect using: COM2

Step 2 Select the serial port at Connect using combobox. Select to C 1 or an C x



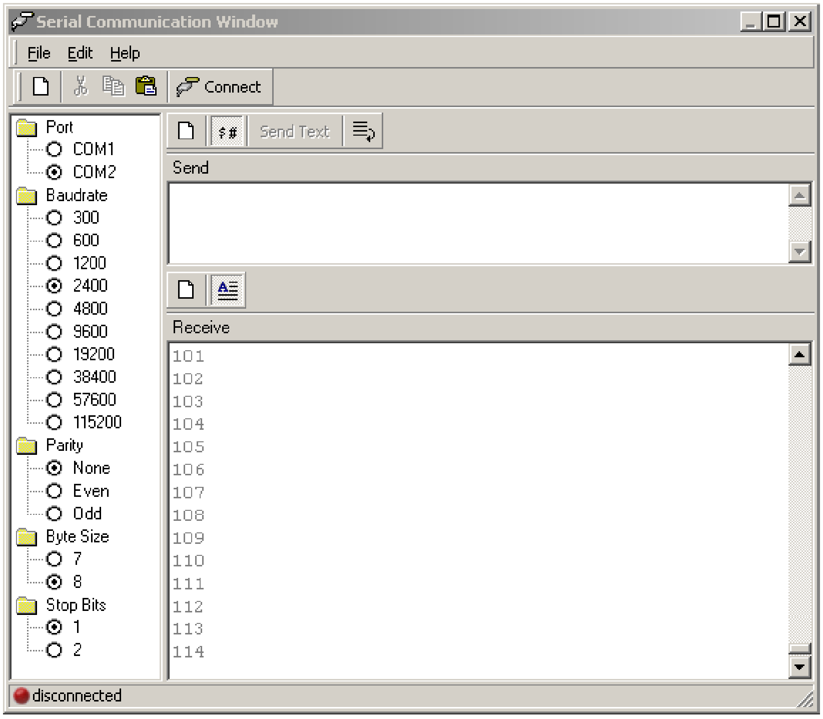
ROBO-PICA2 - HyperTerminal

```
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
```

Disconnected | Auto detect | 2400 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

Step 4 After setting, the communication windows will appear and readf to coonect with microcontroller. The example value in the window is happened after conntection completl .

Figure A1-7 Setting method of Hyper Terminal program



Port
- ○ COM1
- ● COM2

Baudrate
- ○ 300
- ○ 600
- ○ 1200
- ● 2400
- ○ 4800
- ○ 9600
- ○ 19200
- ○ 38400
- ○ 57600
- ○ 115200

Parity
- ● None
- ○ Even
- ○ Odd

Byte Size
- ○ 7
- ● 8

Stop Bits
- ● 1
- ○ 2

```
101
102
103
104
105
106
107
108
109
110
111
112
113
114
```

Figure A1-8 Shows the Serial Communication Window of Microcode Studio

riting program for serial data comminucation of PICBASIC PR compiler

PICBASIC PR compiler provides man commands for serial data communication as

```
SEROUT
SEROUT2
HSEROUT
SERIN
SERIN2
HSERIN
```

SER

SEROUT *Pin*,*Mode*,[*Item* {,*Item...*}]

Pin Transmit port

ode Set the baudrate and data format for communication. Such as 0 means setting baudrate to 2,400 bps or 2 means setting baudrate to ,600 bps

Item Transmit data

Limitation of SER command is the data format fixed to 8 1 cannot change and send onl ASCII code

```
' File : a0106.bas
' Description : Send serial data to computer by SEROUT command
@ DEVICE  PIC16F877,HS_OSC          ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                       ' Use Oscillator 20 MHz
A VAR BYTE
A = 0                               ' Start A with 0
MAIN: SEROUT PORTC.6,2,[A]          ' Send A
      PAUSE 10                      ' Delay 10 Millisecond
      A = A+1                       ' Increment A
      GOTO MAIN                     ' Again
```

Listing A1-6 The demonstration program of SEROUT command operation. This program send ASCII ocde of A variable to computer

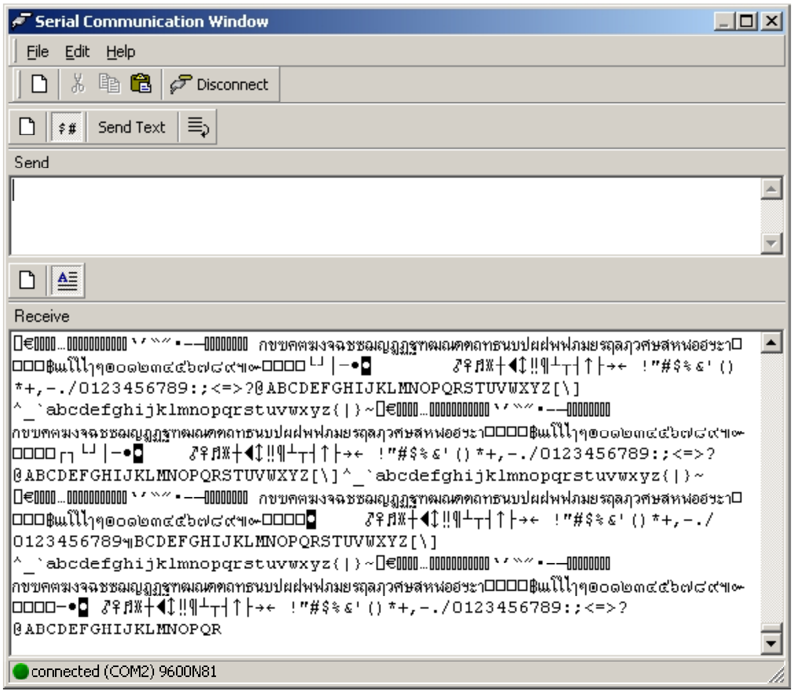
A1.4.1 rite the listing A1-6. Compile and download the code to RBX-877 board.

A1.4.2 Connect RBX-877 board with computer s RS-232 serial port with serial cable.

A1.4.3 pen H per Terminal program or Serial Communication window in icrocode Studio. Select the baudrate to ,600 bit per second. ata format is 8 1

A1.4.4 Run program. bserve the operation of H per Terminal or Serial Communication window in icrocode Studio

The Hyper Terminal shows the operation below



```
□€□□□…□□□□□□□□□□□‘’“”•–—□□□□□□□□ กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรฤลฦวศษสหฬอฮฯะา□
□□□฿เแโใไๅๆ๏๐๑๒๓๔๕๖๗๘๙๚๛□□□□└┘│─•◘          ♂♀♫☼┼◄↕‼¶┴┬┤↑├→←  !"#$%&'()
*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]
^_`abcdefghijklmnopqrstuvwxyz{|}~□€□□□…□□□□□□□□□□□‘’“”•–—□□□□□□□□
กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรฤลฦวศษสหฬอฮฯะา□□□□฿เแโใไๅๆ๏๐๑๒๓๔๕๖๗๘๙๚๛
□□□□┌┐└┘│─•◘       ♂♀♫☼┼◄↕‼¶┴┬┤↑├→←  !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
□€□□□…□□□□□□□□□□□‘’“”•–—□□□□□□□□ กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรฤลฦวศษสหฬอฮฯะา□
□□□฿เแโใไๅๆ๏๐๑๒๓๔๕๖๗๘๙๚๛□□□□◘       ♂♀♫☼┼◄↕‼¶┴┬┤↑├→←  !"#$%&'()*+,-./
0123456789╕BCDEFGHIJKLMNOPQRSTUVWXYZ[\]
^_`abcdefghijklmnopqrstuvwxyz{|}~□€□□□…□□□□□□□□□□□‘’“”•–—□□□□□□□□
กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรฤลฦวศษสหฬอฮฯะา□□□□฿เแโใไๅๆ๏๐๑๒๓๔๕๖๗๘๙๚๛
□□□□─•◘ ♂♀♫☼┼◄↕‼¶┴┬┤↑├→←  !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQR
```

connected (COM2) 9600N81

SER 2

SEROUT2 *DataPin*{\FlowPin},*Mode*,{*Pace*,} {*Timeout*,*Label*,}[*Item...*]

A API Transmit port

E alue to setting baudrate and format data. Such as

If select ,600bps and data format to 8 1. ode value is 84

If select 2,400bps and data format to 8 1. ode value is 3 6

Item Transmit data

The different between SER T and SER T2 command is SER T2 can set the parameter more than SER T command such as data format, transmit format and baudrate. In the listing A1-7 shows the setting data format in decimal (EC) and binar (BI) and shift the cursor to front of line and carriage return b sending ASCII code 0 and 0A.

A1.4.5 rite the listing A1-7. Compile and download the code to RBX-877 board.

A1.4.6 Connect RBX-877 board with computer s RS-232 serial port with serial cable.

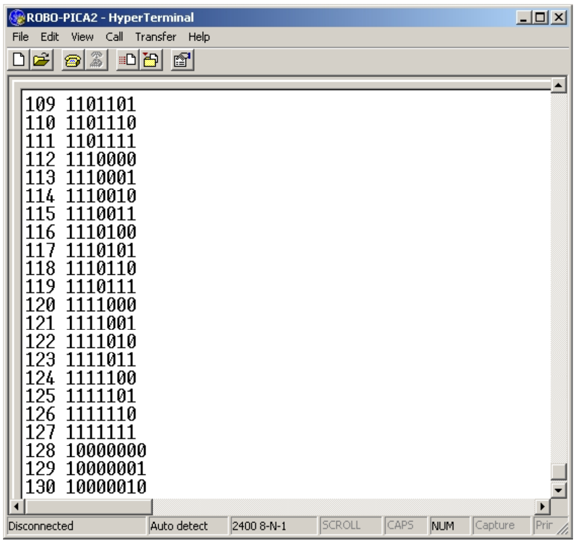
A1.4.7 pen H per Terminal program or Serial Communication window in icrocode Studio. Select the baudrate to 2,400 bit per second. ata format is 8 1

```
' File : a0107.bas
' Description : Send serial data to computer by SEROUT2 command
@ DEVICE  PIC16F877,HS_OSC              ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                           ' Use Oscillator 20 MHz
A    VAR   BYTE
A = 0                                   ' Start A with 0
MAIN: SEROUT2 PORTC.6,396,[DEC A," ",BIN A,$0D,$0A]
                                        ' Show A in Decimal Mode
      PAUSE 10                          ' Delay
      A = A+1                           ' Increment A
      GOTO MAIN
```

Listing A1-7 The demonstration program of SEROUT2 command to sending data to computer

A1.4.8 Run program. bserve the operation of H per Terminal or Serial Communication window in icrocode Studio.

The Hyper Terminal shows the operation below



```
109 1101101
110 1101110
111 1101111
112 1110000
113 1110001
114 1110010
115 1110011
116 1110100
117 1110101
118 1110110
119 1110111
120 1111000
121 1111001
122 1111010
123 1111011
124 1111100
125 1111101
126 1111110
127 1111111
128 10000000
129 10000001
130 10000010
```

SER

HSEROUT [*Item* {,*Item...*}]

Item : Transmit data

HSEROUT is transmit the serial data commnad that use ART or SART module within microcontroller . Before using must set the value to setting the SART module via DEFINE command as

```
DEFINE  HSER_RCSTA  90h      ' Enable receive data
DEFINE  HSER_TXSTA  20h      ' Enable transmit data
DEFINE  HSER_BAUD  2400      ' Set baudrate b  direct
DEFINE  HSER_SPBRG  25       ' Set baudrate via SPBR  register
```

The different between HSER T with SER T2 is HSER T commmand must use with the microcontroller that contains ART or SART module onl and cannot change pin to transmit and receive data. About SER T2 command supports all 14-bit PIC microcontroller (PIC12 6xx, PIC16 6xx 7x 8x 8xx) although an chip do not contain ART or SART module. However HSER T use the memeor space less than SER T2 command

Summar use SER command in PIC microcontroller that support AR or SAR module. Because sa e the code si e more.

```
' File : a0108.bas
' Description : Send serial data to computer by HSEROUT command
@ DEVICE  PIC16F877,HS_OSC          ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                       ' Use Oscillator 20 MHz
DEFINE HSER_RCSTA 90h               ' Set receive register to receiver enabled
DEFINE HSER_TXSTA 20h               ' Set transmit register to transmitter enabled
DEFINE HSER_BAUD 2400               ' Set baud rate
A    VAR         BYTE
A = 0                                                  ' Start A with 0
Main:      HSEROUT ["Innovative "]                     ' Show Text in String
           HSEROUT ["Experiment ",DEC A,$0D,$0A]       ' Show String and Decimal
Number
           A = A+1                                     ' increment A
           GOTO MAIN
```

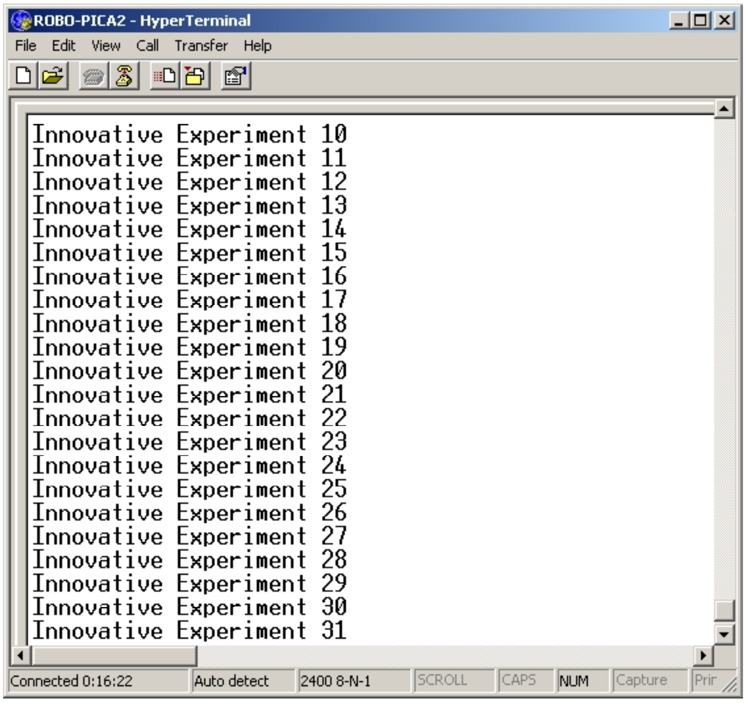
Listing A1-8 The demonstration program of HSEROUT command to sending data to computer

A1.4. rite the listing A1-8. Compile and download the code to RBX-877 board.

A1.4.10 pen H per Terminal program or Serial Communication window in icrocode Studio. Select the baudrate to 2,400 bit per second. ata format is 8 1

A1.4.11 Run program. bserve the operation of H per Terminal or Serial Communication window in icrocode Studio.

The Hyper Terminal shows the operation below



```
Innovative Experiment 10
Innovative Experiment 11
Innovative Experiment 12
Innovative Experiment 13
Innovative Experiment 14
Innovative Experiment 15
Innovative Experiment 16
Innovative Experiment 17
Innovative Experiment 18
Innovative Experiment 19
Innovative Experiment 20
Innovative Experiment 21
Innovative Experiment 22
Innovative Experiment 23
Innovative Experiment 24
Innovative Experiment 25
Innovative Experiment 26
Innovative Experiment 27
Innovative Experiment 28
Innovative Experiment 29
Innovative Experiment 30
Innovative Experiment 31
```

SERI

SERIN *Pin*,*Mode*,{*Timeout*,*Label*,}{[*Qual...*],}{*Item...*}

Pin Receive port. or RBX-877 board set to RC7 (P RTC.7)

ode Set the baudrate and data format for communication. Such as 0 means setting baudrate to 2,400 bps or 2 means setting baudrate to ,600 bps

imeout aiting receive time in millisecond unit. If time over, CP will jump to the destination that define b **Label**

ual Comparison data. If receive data match with this data, load next data into **Item ariable**

SERI command is easiest of recei e data from computer ia serial port. his command cannot check an communication status. It checks onl the waiting time.

A1.4.12 rite the listing A1- . Compile and download the code to RBX-877 board.

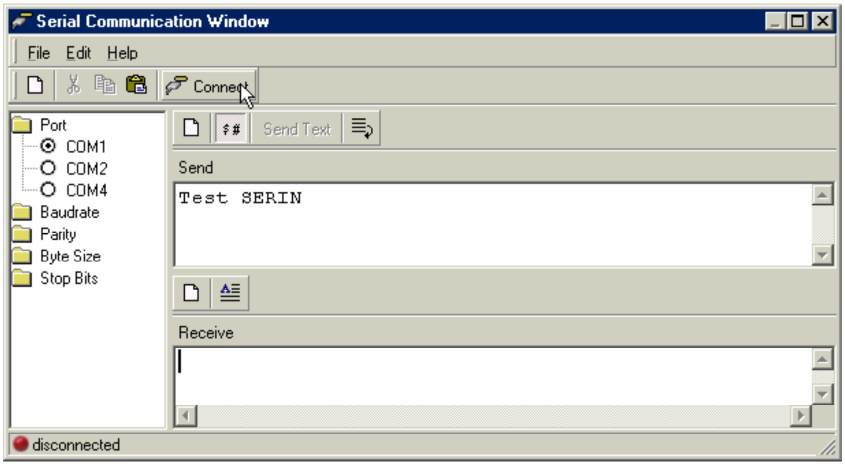
```
' File : a0109.bas
' Description : Receive serial data from computer by SERIN command
@ DEVICE  PIC16F877,HS_OSC          ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                       ' Use Oscillator 20 MHz
DEFINE     LCD_DREG    PORTD        ' Set LCD Data port
DEFINE     LCD_DBIT    4            ' Set starting Data bit (0 or 4) if 4-bit
bus
DEFINE     LCD_RSREG   PORTC        ' Set LCD Register Select port
DEFINE     LCD_RSBIT   0            ' Set LCD Register Select bit
DEFINE     LCD_EREG    PORTC        ' Set LCD Enable port
DEFINE     LCD_EBIT    5            ' Set LCD Enable bit
DEFINE     LCD_BITS    4            ' Set LCD bus size (4 or 8 bits)
DEFINE     LCD_LINES   2            ' Set number of lines on LCD
DEFINE     LCD_COMMANDUS 2000       ' Set command delay time in us
DEFINE     LCD_DATAUS  50           ' Set data delay time in us
A    VAR         BYTE
LOOP:      SERIN PORTC.7,0,["R"],A
           LCDOUT $FE,$01,A
           GOTO LOOP
```

Listing A1-9 The demonstration program of SERIN command to receive the serail data from computer and show at LCD module on RBX-877 board

A1.4.13 pen H per Terminal program or Serial Communication window in icrocode Studio. Select the baudrate to 2,400 bit per second. ata format is 8 1. However for getting the best result from this experiment should be use Serial Communication window in icrocode Studio better. Because this window can show the transmit data from computer.

A1.4.14 Run program. Press ke board to send data to RBX-877 via serial port b using Send box in Serial Comminication window of icrocode Studio. Then click Connect button. bserve the LC odule operation compare with the result at Serial Communication window of icrocode Studio

The Serial Comm nication of Microcode St dio shows the operation below



Send: Test SERIN

SERI 2

SERI 2 command is developed fro SERI command. This command can select man data format and check the receiving data. Its s stax is

SERIN2 *DataPin*{\FlowPin},*Mode*,{*ParityLabel*,} {*Timeout*,*Label*,}[*Item...*]

ataPin Receive port. or RBX-877 board set to RC7 (P RTC.7)

ode alue to setting baudrate and format data. Such as

If select ,600bps and data format to 8 1. ode value is 84

If select 2,400bps and data format to 8 1. ode value is 3 6

Parit Label Position to jump if found the parit checking error. Enable b setting bit 13 of ode parameter

imeout aiting receive time in millisecond unit. If time over, CP will jump to the destination that define b **Label**

Item It means a variable or more for saving data that received from atapin and can set variable t pe to arra for storing data man b tes.

```
' File : a0110.bas
' Description : Receive serial data from computer by SERIN2 command
@ DEVICE  PIC16F877,HS_OSC          ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                       ' Use Oscillator 20 MHz
DEFINE     LCD_DREG    PORTD        ' Set LCD Data port
DEFINE     LCD_DBIT    4            ' Set data bit start at bit4 in 4-bit mode
DEFINE     LCD_RSREG   PORTC        ' Set LCD Register Select port
DEFINE     LCD_RSBIT   0            ' Set LCD Register Select bit
DEFINE     LCD_EREG    PORTC        ' Set LCD Enable port
DEFINE     LCD_EBIT    5            ' Set LCD Enable bit
DEFINE     LCD_BITS    4            ' Set LCD bus size (4 or 8 bits)
DEFINE     LCD_LINES   2            ' Set number of lines on LCD
DEFINE     LCD_COMMANDUS     2000   ' Set command delay time in us
DEFINE     LCD_DATAUS  50           ' Set data delay time in us
DAT  VAR         BYTE[9]
LOOP:   SERIN2 PORTC.7,396,3000,OUT,[str DAT\9 ]
                                    ' Recieve Data 9 bytes save to DAT
           LCDOUT $FE,$01,STR DAT\9       ' Show data on LCD Module
           GOTO LOOP                      ' Again
OUT:       FREQOUT PORTA.4,100,2000       ' If Timeout, generate sound
           GOTO LOOP
```

Listing A1-10 The demonstration program of SERIN2 command to receive the serail data from computer and show at LCD module on RBX-877 board

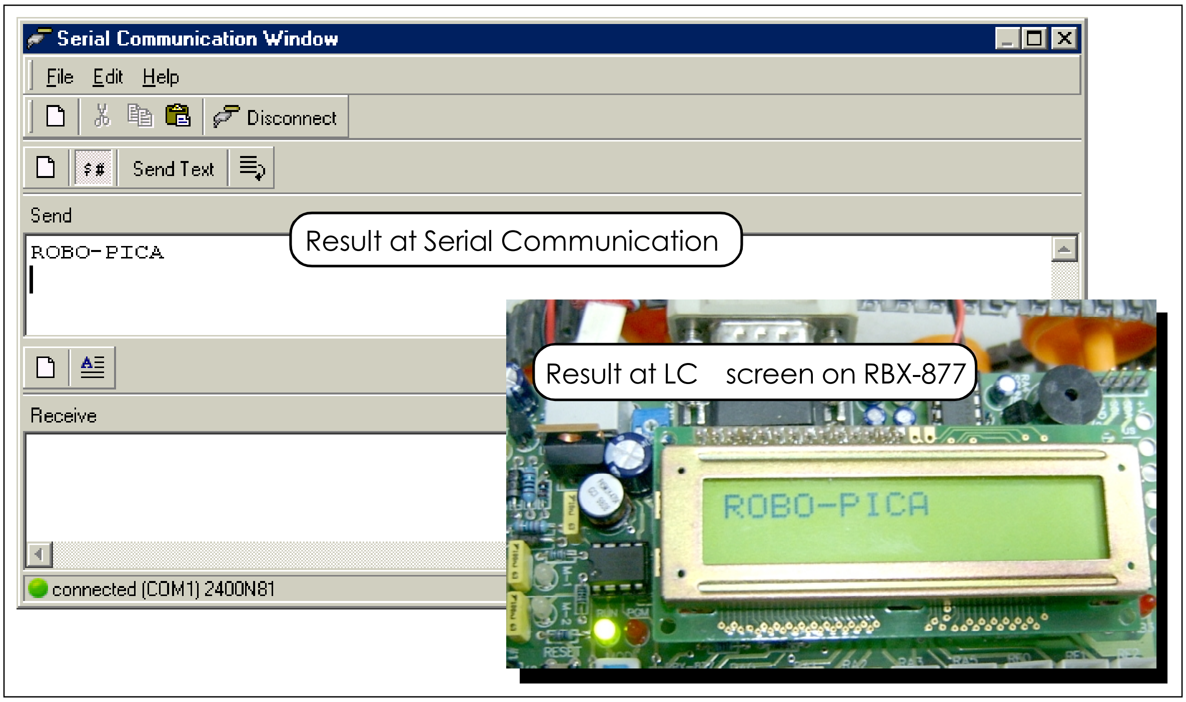
A1.4.15 rite the listing A1-10. Compile and download the code to RBX-877 board.

A1.4.16 Connect RBX-877 board with computer s RS-232 serial port with serial cable.

A1.4.17 pen H per Terminal program or Serial Communication window in icrocode Studio. Select the baudrate to 2,400 bit per second. ata format is 8 1. However for getting the best result from this experiment should be use Serial Communication window in icrocode Studio better. Because this window can show the transmit data from computer.

A1.4.18 Run program. Press ke board to send data to RBX-877 via serial port b using Send box in Serial Comminication window of icrocode Studio. Then click Connect button. bserve the LC odule operation compare with the result at Serial Communication window of icrocode Studio

The Serial Comm nication of Microcode St dio shows the operation in fig re A1-11 and see the res lt at C screen



ROBO-PICA

connected (COM1) 2400N81

Result at Serial Communication

Result at LC   screen on RBX-877

Figure A1-9 Shows the result of the listing A-10 operation at Serial communication window of Microcode Studio and LCD module on RBX-877 board

SERI

`HSEROUT` is receive the serial data command that use   ART or   SART module within microcontroller . Before using must set the value to setting the   SART module via `DEFINE` command as

```
DEFINE  HSER_RCSTA  90h     ' Enable receive data
DEFINE  HSER_TXSTA  20h     ' Enable transmit data
DEFINE  HSER_BAUD  2400     ' Set baudrate b   direct
DEFINE  HSER_SPBRG  25      ' Set baudrate via SPBR    register
```

Receiver port pin is fixed to RxD pin of UART module. For PIC16F877 microcontroller is RC7. The syntax of this command as :

HSERIN `{ParityLabel,}{Timeout,Label,}[Item{,...}]`

Parit Label   Position to jump if found the parit   checking error.

imeout   aiting receive time in millisecond unit. If time over, CP   will jump to the destination that define b   **Label**

Item   It means a variable or more for saving data that received from RC7 pin and can set variable t pe to arra   for storing data man   b tes.

The different between `HSERIN` with `SERIN2` is `HSERIN` commmand must use with the microcontroller that contains ART or SART module onl and cannot change pin to transmit and receive data. About `SERIN2` command supports all 14-bit PIC microcontroller (PIC12 6xx, PIC16 6xx 7x 8x 8xx) although an chip do not contain ART or SART module. However `HSERIN` use the memeor space less than `SERIN2` command

Summar use SERI command in PIC microcontroller that support AR or SAR module. Because sa e the code si e more.

A1.4.1 rite the listing A1-11. Compile and download the code to RBX-877 board.

A1.4.20 Connect RBX-877 board with computer s RS-232 serial port with serial cable.

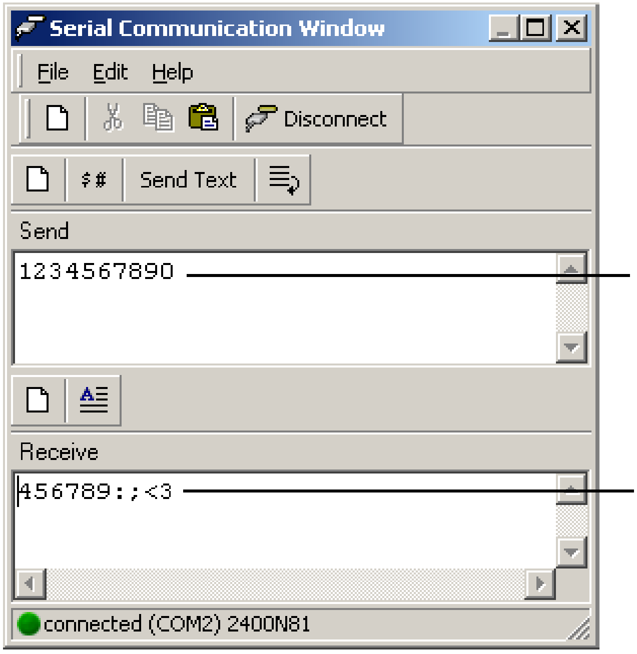
A1.4.21 pen H per Terminal program or Serial Communication window in icrocode Studio. Select the baudrate to 2,400 bit per second. ata format is 8 1. However for getting the best result from this experiment should be use Serial Communication window in icrocode Studio better. Because this window can show the transmit data from computer.

```
' File : a0111.bas
' Description : Receive serial data from computer by HSERIN command
@ DEVICE  PIC16F877,HS_OSC    ' Use PIC16F877 and HS Oscilator
DEFINE OSC 20                 ' Use Oscillator 20 MHz
DEFINE HSER_RCSTA 90h         ' Set receive register to receiver enabled
DEFINE HSER_TXSTA 20h         ' Set transmit register to transmitter enabled
DEFINE HSER_BAUD 2400         ' Set baud rate
DAT  VAR    BYTE

LOOP: HSERIN [DAT]            ' Recieve Data 9 Byte Save To DAT
      DAT = DAT + 3
      HSEROUT [DAT]           ' Send Back To Serial
      GOTO LOOP               ' Again
```

Listing A1-11 The demonstration program of HSERIN command to receive the serial data from computer and modify with math routine then send back to computer for display on the Serial Communication window of Microcode Studio

A1.4.22 Run program. Press ke board to send data to RBX-877 via serial port b using Send box in Serial Comminication window of icrocode Studio. Then click Connect button. bserve the result at Serial Communication window of icrocode Studio.

The Serial Comm nication of Microcode St dio shows the operation below :



Send: 1234567890

Receive: 456789:;<3

connected (COM2) 2400N81

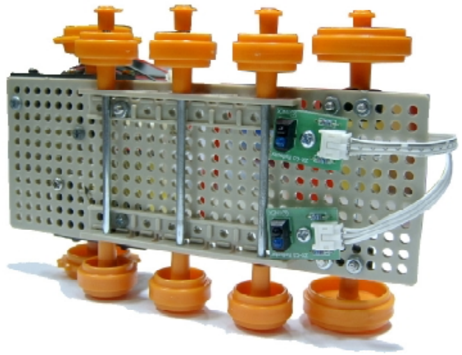
Transmit data

Receive data It s e ual value of transmit data 3. or example. transmit ASCII code of number 1 (0x21). The receive data will be 0x24. It means ASCII code of number 4

Chapter 3

Building Robo-PICA kit

This chapter describes about how to building the Robo-PICA robot kit. The robot in this chapter is install the basic sensor, ZX-03 Infrared reflector. The features of Robo-PICA robot kit are :



● Driving with DC motor gearboxes and Track wheel

● Controlled by PIC16F877 microcontroller

● 8KWords program memory

● Re-programmable at least 10,000 times for flash program memory

● Support many types of sensor and detectors such as

ZX-01 Switch input board for attacking detection,

ZX-03 Infrared Reflector for line tracking and area,

ZX-05 Infrared receiver module for remote controlling,

ZX-08 Infrared Objector for contactless object avoiding,

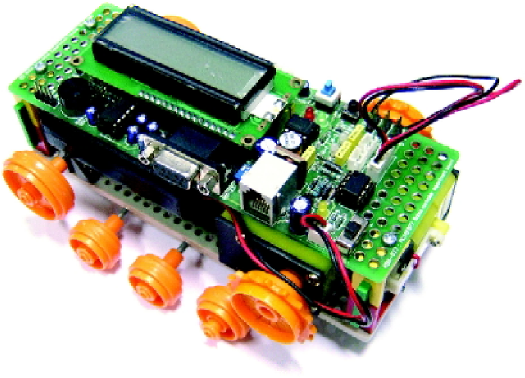
GP2D120 Infrared distance sensor,

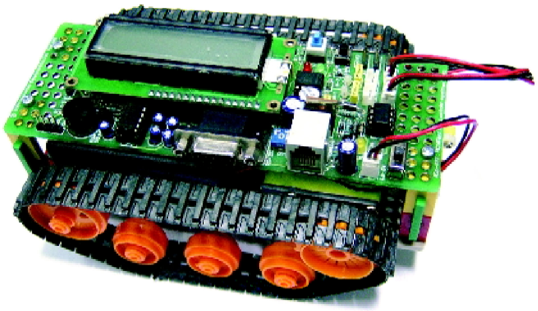
SRF05 Ultrasonic sensor,

CMPS03 Digital compass,

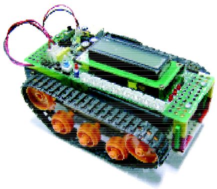
Memsic2125 Accelerometer sensor

and more...





● Provides Character LCD moduel 16x2 and LED status for displaying thre robot operation.



Activity 2

Make the Robo-PICA

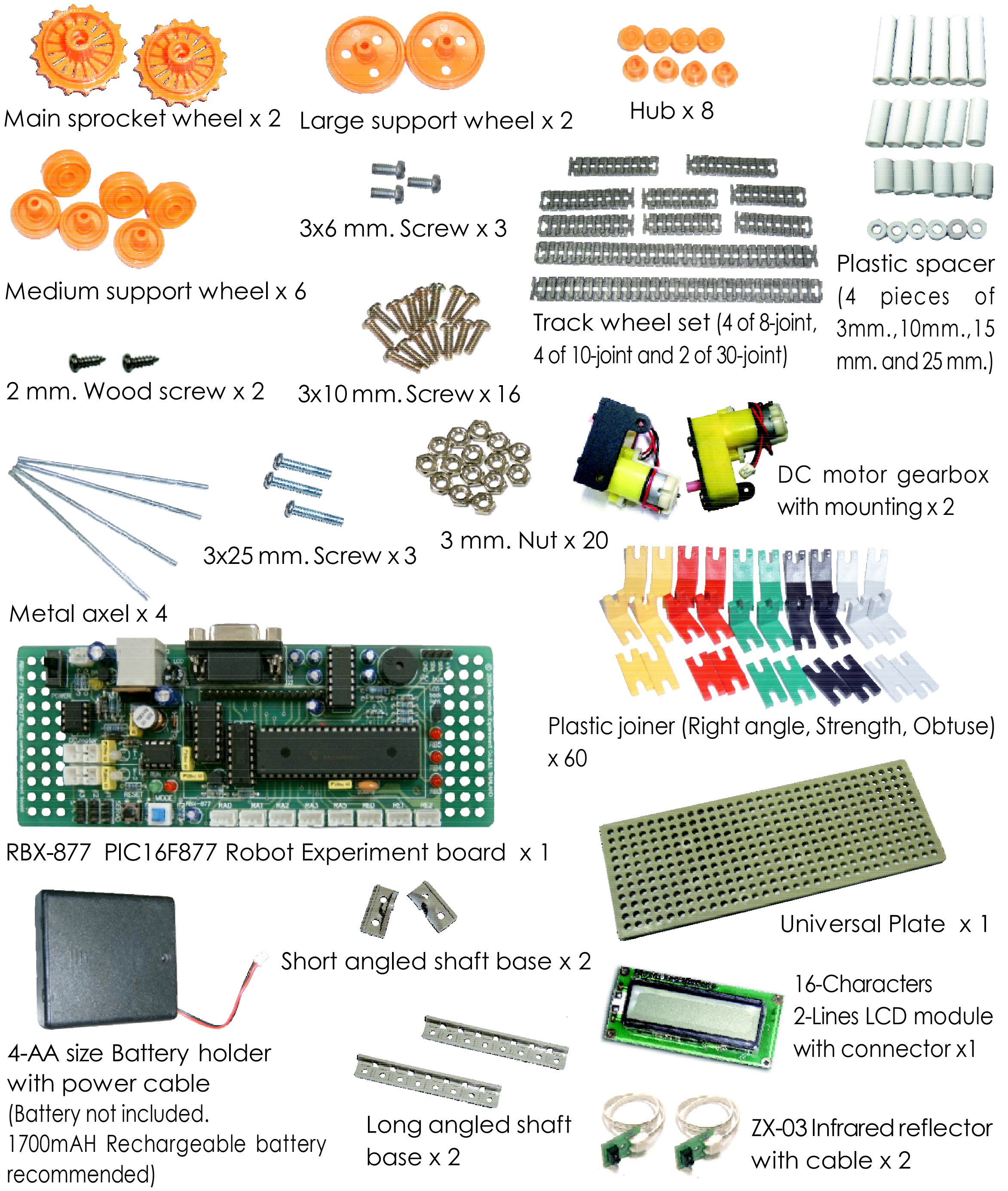
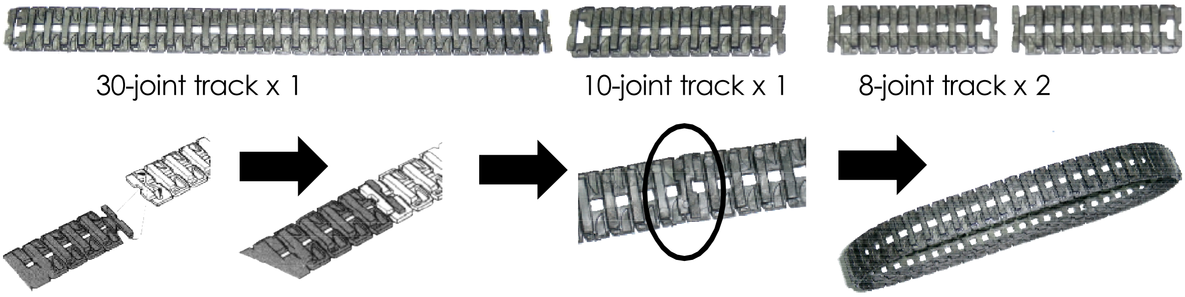

Main sprocket wheel x 2
Large support wheel x 2
Hub x 8
Medium support wheel x 6
3x6 mm. Screw x 3
Track wheel set (4 of 8-joint, 4 of 10-joint and 2 of 30-joint)
Plastic spacer (4 pieces of 3mm., 10mm., 15 mm. and 25 mm.)
2 mm. Wood screw x 2
3x10 mm. Screw x 16
DC motor gearbox with mounting x 2
Metal axel x 4
3x25 mm. Screw x 3
3 mm. Nut x 20
Plastic joiner (Right angle, Strength, Obtuse) x 60
RBX-877 PIC16F877 Robot Experiment board x 1
Universal Plate x 1
Short angled shaft base x 2
4-AA size Battery holder with power cable (Battery not included. 1700mAH Rechargeable battery recommended)
16-Characters 2-Lines LCD module with connector x1
Long angled shaft base x 2
ZX-03 Infrared reflector with cable x 2

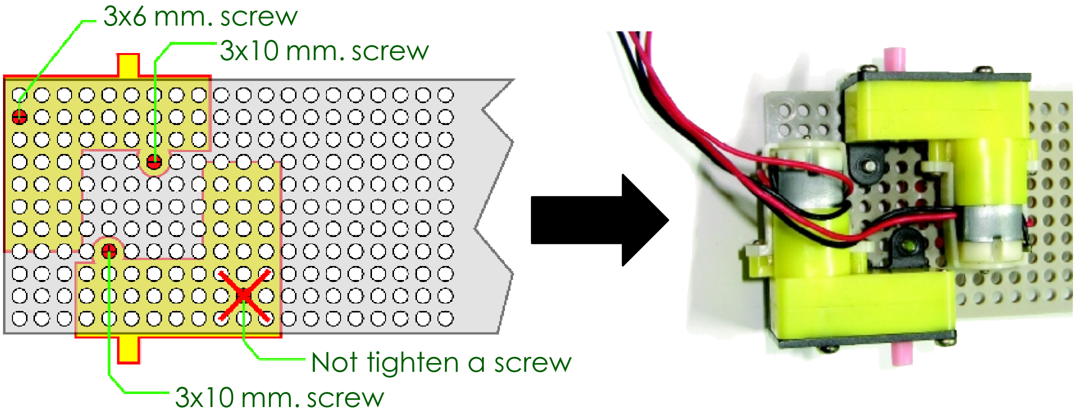
Figure A2-1 Shows all parts of Robo-PICA robot kit

A2.1 Create two track belts by putting the different size tracks together. One track would consist of the following: One 30-joint track, one 10-joint track, and two 8-joint track. Connect the 10-joint track to the end of the 30-joint track. Next connect the two 8-joint tracks together. Take one end and connect it to the other end of the 10-joint track. Then take the other end of the 8-joint track and connect it to the remaining end of the 30-joint track to form one complete loop. Repeat the steps to make two track sets. If the track is too tight or loose, you can either adjust the length of the track or adjust the position of the short angled shaft base until the track has a good fit.

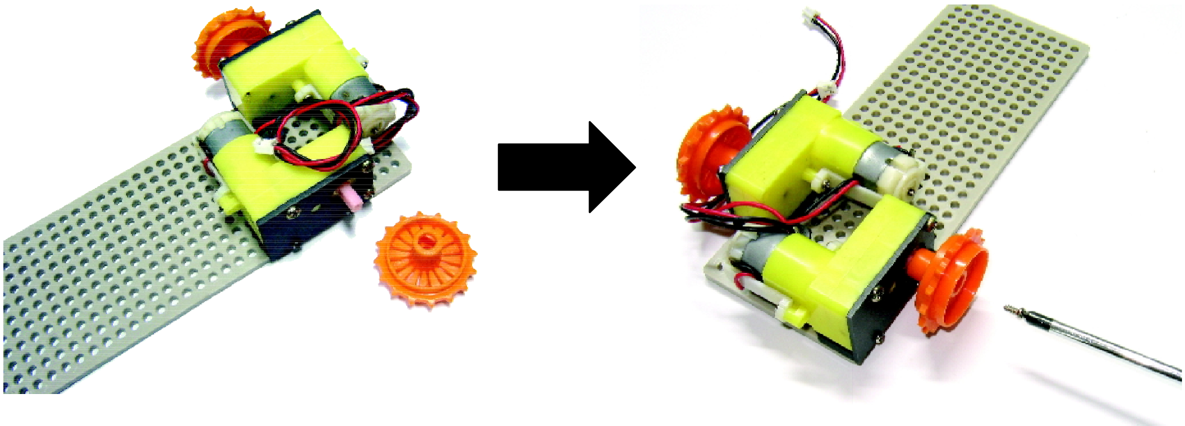


Buiding the track wheel from many tracks connection has purpose to builders can change the length of wheel by their own and the robot does not nescessory same size from example.

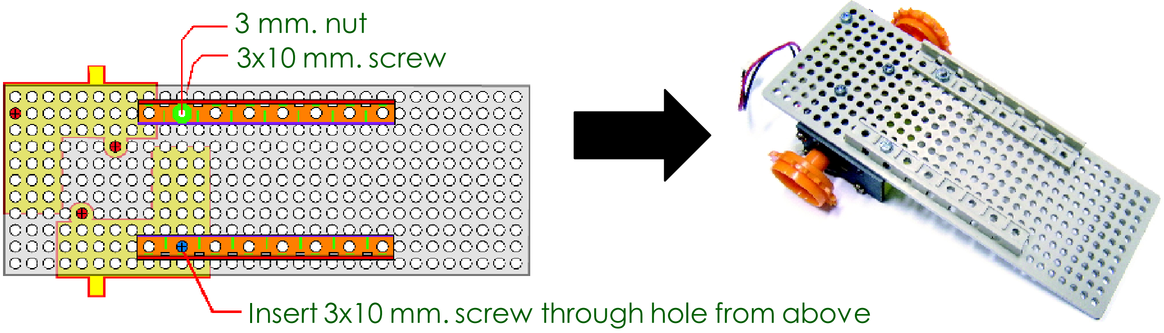
A2.2 Attach the motor gearbox sets to the universal plate using a 3 x 6 mm. and 3 x 10 mm. screw. Position the Motor Gearbox sets as shown in the picture below. Attach only 3 points because 1 point remainder will be use for attach a Long angled shaft base.



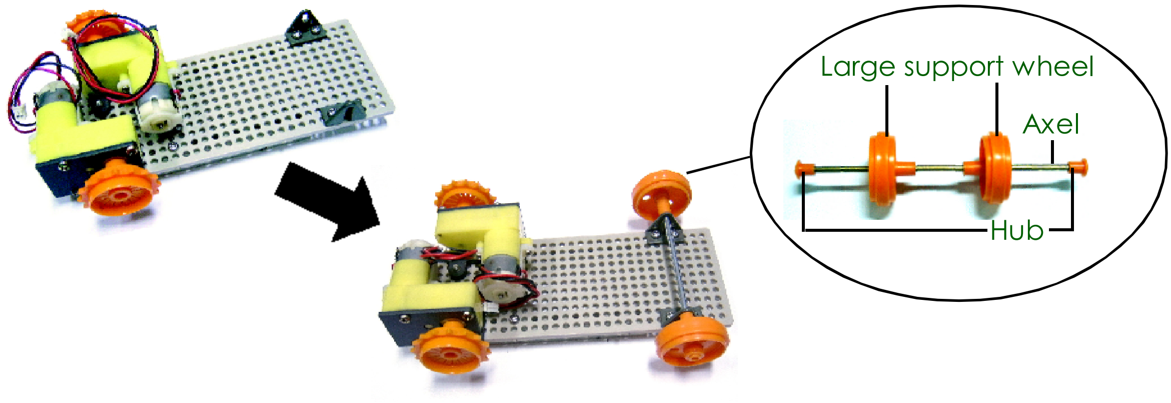
A2.3 Insert the Main Sprocket Wheel into the motor gearbox and screw it in tightly with a 2 mm. Wood screw.



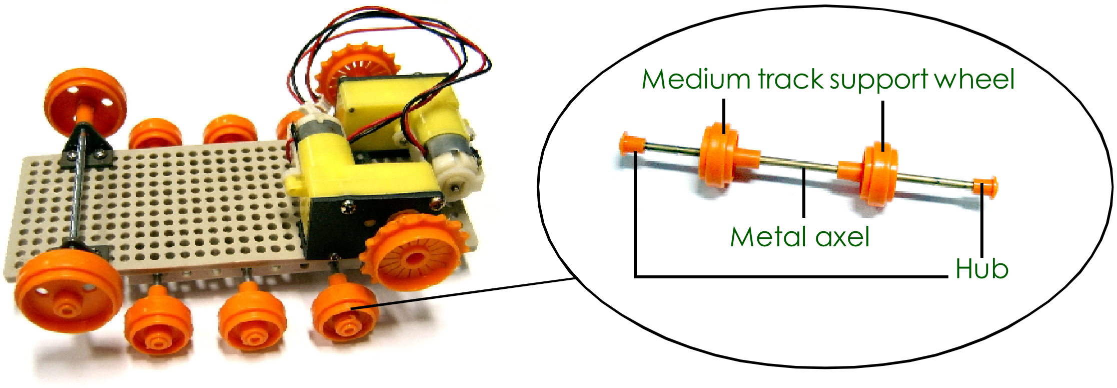
A2.4 Attach the Long angled shaft base with the plate by 3x10 mm. Screw. One position tighten a screw into the remainder hole from step A2.2. Another end, insert 3x10 mm. Screw from above through the plate's hole and tighten by 3 mm. Nut. Attach another Long angled shaft base in same method.



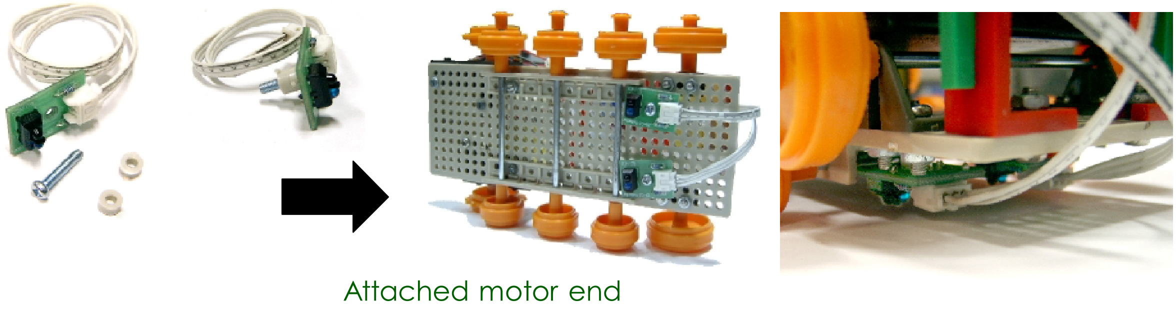
A2.5 Attach the short angled shaft base at the end of the plate (opposite the motor attached end) as shown in the picture by counting the hole from the end 4 holes. Use 3x10 mm. Screw and 3 mm. Nut to tighten the short angled shaft base. Make them both side. Next, insert the metal axel through the hole of the short angled shaft base. Insert two Large support wheels to both end of axel and close by Hubs



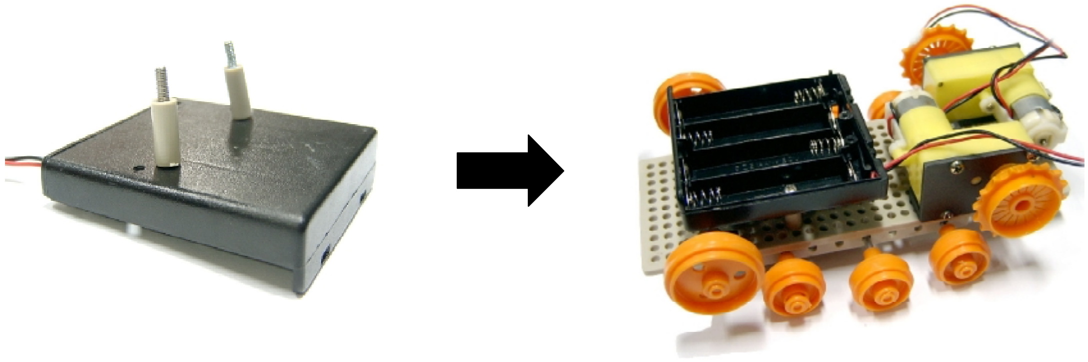
A2.6 Insert the metal axel into the holes of the long angled shaft in the hole positions of 1, 4, and 7 (Counting from any side). Place the Medium track support wheels over the metal axel as shown in the pictures. Insert the hubs over the wheels so that the wheels and the axels are connected tightly.



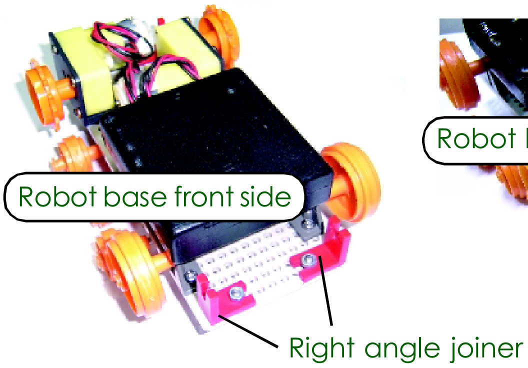
A2.7 Insert 3x15 mm. Screw through the ZX-03 Infrared reflector and 2 of 3 mm. Plastic spacer. Make 2 sets. Next, attach both sensors with the front of bottom base (opposite the attached motor end). Distance between both sensor is about 3 to 4 cm. Tighten with 3 mm. Nuts. The sensor will far from the floor about 5 mm.

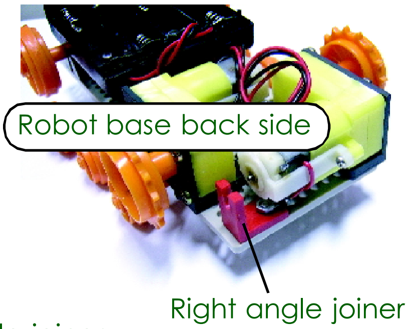


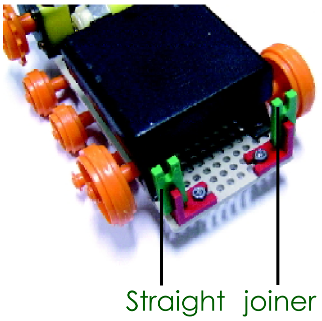
A2.8 Open the cover of Battery holder. Insert 2 of 3x25 mm. Screws through the Battert holder and insert 15 mm. Plastic spacer via both screws. Attach the Battery holder with the Robot base. Turn the cable end in same direction with motor. Brecause the motor output connector abd battery is near. Tighten by 3 mm. Nuts.



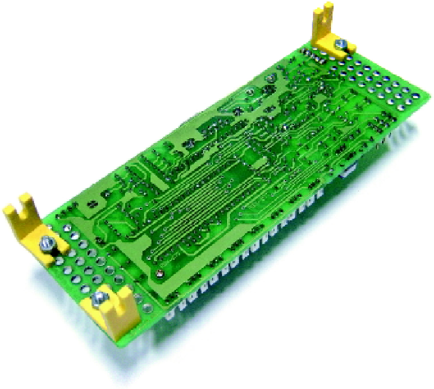
A2.9 Attach 2 of Right angle joiner at front position of robot base with 3x10 mm. Screws and 3 mm. Nuts. Tighten all together.At the back end, attach a Right angle joiner at the corner of Robot base. Al l 3-position ofRight angle joiner will relate with the attachment holes of RBX-877 board. Connect the Strength joiner at the end of 3 Right angle joiner for support the RBX-877 board attachment.





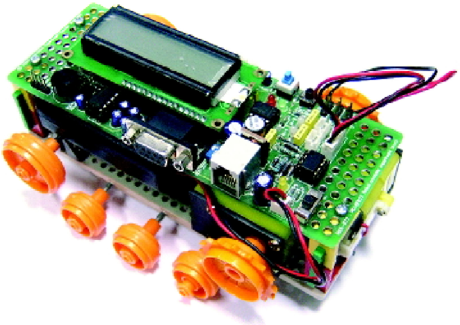


A2.10 Attache 3 of Right angle joiners into the ERBX-877 board at the straight position from the Right angle joiner attachment from step A2.9 by 3x10 mm. Screws and 3 mm. Nuts.



A2.11 Load 4 of AA size batteries into the Battery holder and close the cover. Coonect the RBX-877 board that attached Right angle joiners with the Robot base from step A2.9.With this technique, builders can remove the board for chamging the batteries easier.

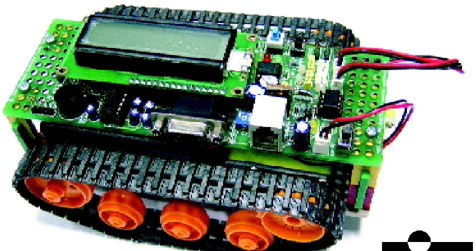
A2.12 Plug the motor cable to M-1 and M-2 connector on RBX-877 board. The connection can change for adjust the motor direction to correct in the movement activity. Plug power cable from the Battery holder to Battery connector on RBX-877 board. Keep all cables be neat and tidy.



A2.13 Attach the tracks to the supporting wheels of the robot.

A2.14 Plug ZX-03 Infrared Reflector's cable to RA0 and RA1 of RBX-877 board.

Now..Robo-PICA ready to program and Run it.



Chapter 4

Simple robot 's programming control

The first thing to control robot is ovement control. The heart of movement is C motor circuit In Robo-PICA use C motor gearbox in driving. The figure 4-1 shows the C motor circuit. PIC16 877 assigns 6 port pins to connect the C motor driver circuit for driving 2 motors.

otor driver format has 4 t pes as

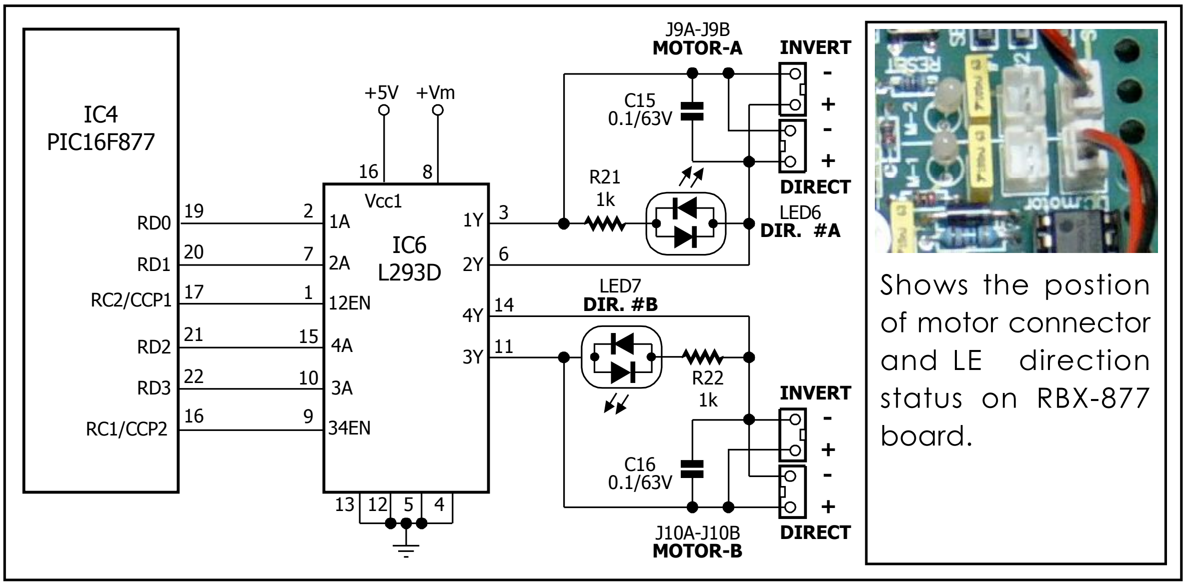
(1) Clockwise motor driving

(2) Anti-clockwies motor driving

(3) otor s shaft is free

(4) otor s shaft is locked or Brake

The heart of C motor driver circuit is L2 3 H-Bridge driver (ma be use S 754410 replacement). In the table 4-1 shows all signals to control the C motor driver circuit.



Shows the postion of motor connector and LE direction status on RBX-877 board.

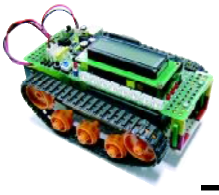
Figure 4-1 The DC motor driver schematic of RBX-877 board

12EN 34EN pin	1A 3A pin	2A 4A pin	M t p ti n
0	X	X	Shaft free
1	0	0	Shaft locked or Brake
1	0	1	Clockwise turning
1	1	0	anti-clockwise turning
1	1	1	Shaft locked or Brake

X means logic "0" or "1"

e 4-1 shows logic signal to control motor direction

L2 3 outputs connect to C motor gearbox and provides LE status for motor suppl voltage. If suppl in IRECT, LE will lights in reen. The opposite, if red LE lights, it means suppl voltage is I ERT. Buiders can use the different color for defining direction. In other same meaning, if red LE turned on, the robot will be backwards movements. If green LE turned on, the robot will be forwards movements.



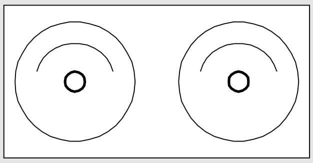
Acti it 3

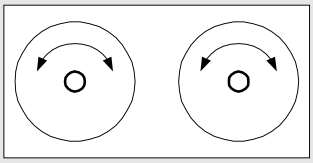
Check motor rotation

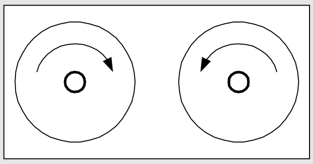
A3.1 pen icrocode Studio. rite the BASIC code following the listing P3-1.

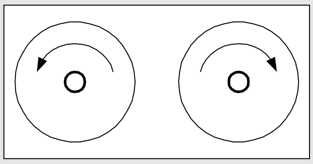
A3.2 Connect the download cable to the robot. Turn-on power. Select E switch to program mode (red LE at P . label lights). Click button or press 10 ke for compile and download program to Robo-PICA.

```
'
'   Name    : P0301.BAS
'   Notes   : Test Direction of Motor
'
@ DEVICE PIC16F877,HS_OSC          used PIC16F877
DEFINE  OSC 20                     Oscillator = 20 MHz
LOOP:
       Motor Free
      FREQOUT PORTA.4,200,2000     Beep
 ..... Motor A .........
      LO     PORTC.2               Disable Motor A
      LO     PORTD.0
      LO     PORTD.1
 ..... Motor B .........
      LO     PORTC.1               Disable Motor B
      LO     PORTD.2
      LO     PORTD.3
      PAUSE 5000                   Delay 5 Second

       Motor Lock
      FREQOUT PORTA.4,200,2000     Beep
 ..... Motor A .........
      HIGH   PORTC.2               Enable Motor A
 ..... Motor B .........
      HIGH   PORTC.1               Enable Motor B
      PAUSE  5000                  Delay 5 Second

       Motor Forward
    FREQOUT PORTA.4,200,2000       Beep
 ..... Motor A .........
      LO     PORTD.0
      HIGH   PORTD.1
 ..... Motor B .........
      HIGH   PORTD.2
      LO     PORTD.3
      PAUSE 5000                   Delay 5 Second

       Motor Backword
      FREQOUT PORTA.4,200,2000     Beep
 ..... Motor A .........
      HIGH   PORTD.0
      LO     PORTD.1
 ..... Motor B .........
      LO     PORTD.2
      HIGH   PORTD.3
      PAUSE  5000                  Delay 5 Second
      GOTO   LOOP
```









Listing 3-1 The driving motor demonstration program of Robo-PICA

A3.3 Select to R   mode (green LE at R   label lights). bserve the operation.

First step efine all motor control port with logic 0 . o voltage appl to both motor. otor shafts will be free.

Second step Enable both Cc motor driver circuits b send logic 1 to enable port pin. But motor port remain as logic 0 . otor pole receive negative voltage both. The motor shaft will be locked.

hird step Control motor to move forward b appl positive voltage to one of motor pole. R 0 and R 2 still get logic 1 . Another pole appl 0 becasue R 1 and R 3 port as logic 0

Fourth step Change motor direction b swap the logic appl to motor pole. R 0 and R 2 port pin as logic 0 but R 1 and R 3 port pin as logic 1 .

Addition in ever changing step, icrocontroller will drive sound 2kHz fre uenc 0.2 second for reporting the builders.

hen running....See the robot moving and LE status operation.

In for ard mo ement, both motor LE status must show green color.

In ac   ard, chagne to red color both.

If not, must change the direction of motor cable conection to RBX-877 board. Change until the operation correct.

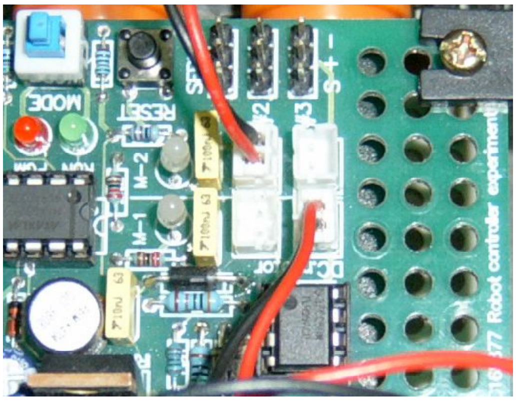
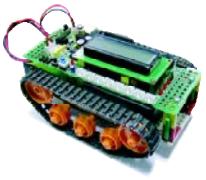


Figure A3-1 Changing the motor cable connection is nescessary if the movement and motor's LED status is not same.

In low power C motor operation altho gh apply the signal to lock the motor s shaft sometime cannot lock beca se internal resistance is very low



Acti it 4

Simple mo ement control of Robo-PICA

Robo-PICA moves forward or backward b driving both C motor gearbox in same direction and same time. If need to turn or rotate, this is method

1. **Stop one motor and ri e another one** If stop left motor and drive right motor,the robot will turn left. In the opposite, stop right motor and drive left motor. The robot will turn right. The speed will similar forward moving but motor must take more tor ue because drive the track wheel at the stop motor. The turning point of this moving is sta at the stop wheel. *See fig re* A4-1

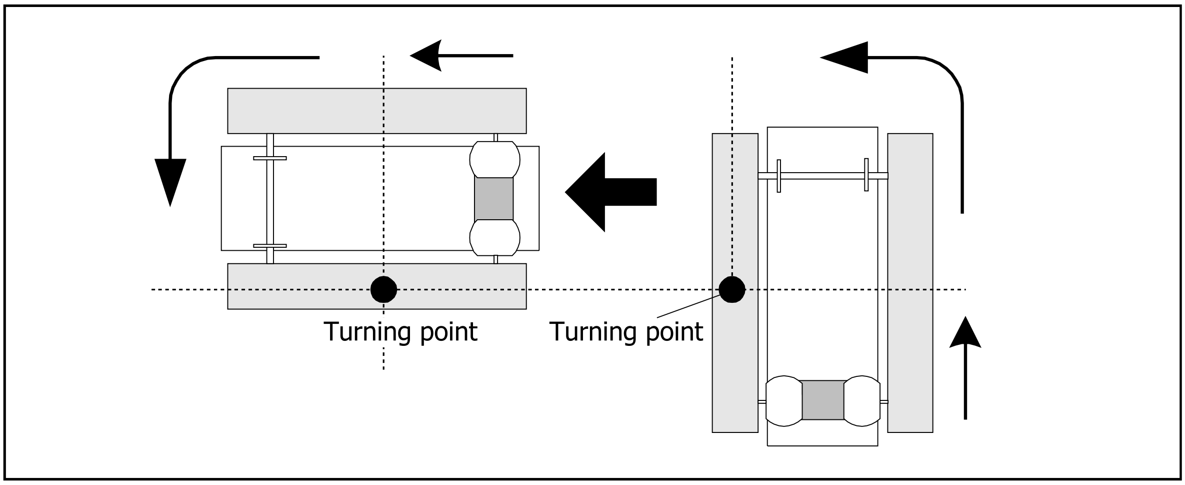


Figure A4-1 Turning method of Robo-PICA by stop a motor and fix a wheel.

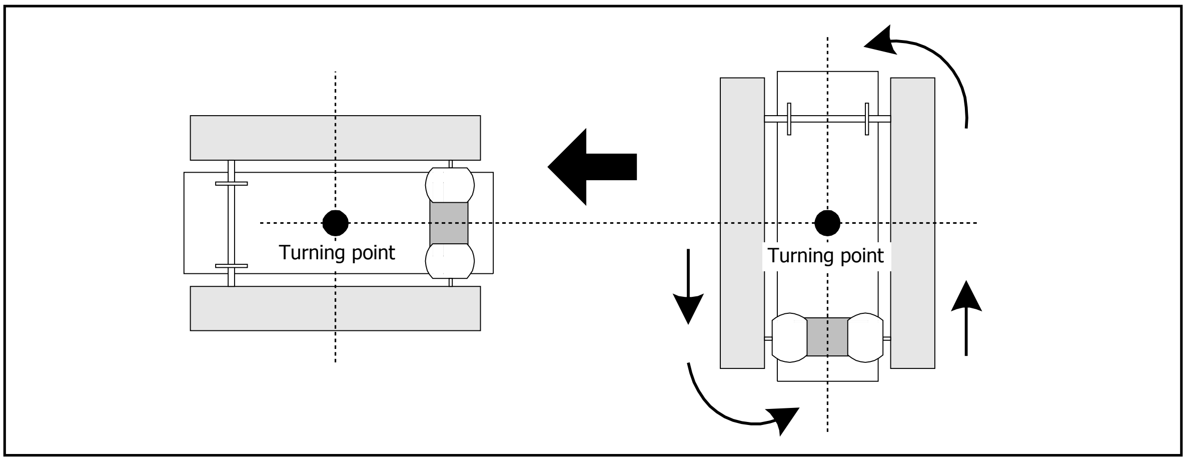


Figure A4-2 Turning method of Robo-PICA by driving both motors in opposite direction.

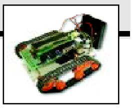
2. **ri e both motors in opposite direction** If the left motor drives forward and right motor drives backward, the robot will rotate right direction. If opposite, left motor drives backward and right motor drives forward. The robot will rotate left direction instead. In this method the speed of rotastion will be increase 2 times. and less fiction. The turning point is center of robot bod . See figure A4-2.

A4.1 rite the BASIC code following the listing P4-1. Connect the download cable to the robot. Turn-on power. Select E switch to program mode. Click button or press 10 ke for compile and download program to Robo-PICA.

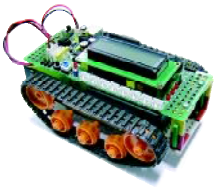
A4.2 Select to R mode (green LE at R label lights). bserve the operation.

The robot will t rn left 5 seconds t rn right 5 seconds rotate left 5 seconds and rotate right 5 seconds Speed movement in rotatation will be faster than simple t rning ewvery change movement the beep so nd is driven

```
'
'   Name     : P0401.BAS
'   Notes    : Turn Robot And Spin Robot
'
@ DEVICE PIC16F877,HS_OSC                    used PIC16F877
DEFINE  OSC 20                               Oscillator = 20 MHz
HIGH PORTC.2                                 Enable Motor A
HIGH PORTC.1                                 Enable Motor B

LOOP:
       Motor Turn Left
     FREQOUT PORTA.4,200,2000                Beep
     LO    PORTD.0 :   LO    PORTD.1        .. Motor A ....
     HIGH  PORTD.2 :   LO   PORTD.3         .. Motor B ....
     PAUSE 5000                              Delay 5 Second
       Motor Turn Right
     FREQOUT PORTA.4,200,2000                Beep
     LO    PORTD.0 :   HIGH  PORTD.1        .. Motor A ....
     LO    PORTD.2 :   LO   PORTD.3         .. Motor B ....
     PAUSE 5000                              Delay 5 Second
       Motor Spin Left
     FREQOUT PORTA.4,200,2000                Beep
     HIGH  PORTD.0 :   LO    PORTD.1        .. Motor A ....
     HIGH  PORTD.2 :   LO   PORTD.3         .. Motor B ....
     PAUSE 5000                              Delay 5 Second
       Motor Spin Right
     FREQOUT PORTA.4,200,2000                Beep
     LO    PORTD.0 :   HIGH  PORTD.1        .. Motor A ....
     LO    PORTD.2 :   HIGH  PORTD.3        .. Motor B ....
     PAUSE 5000                              Delay 5 Second
     GOTO LOOP
```



Listing 4-1 The turning and rotation demonstrsation program for Robo-PICA



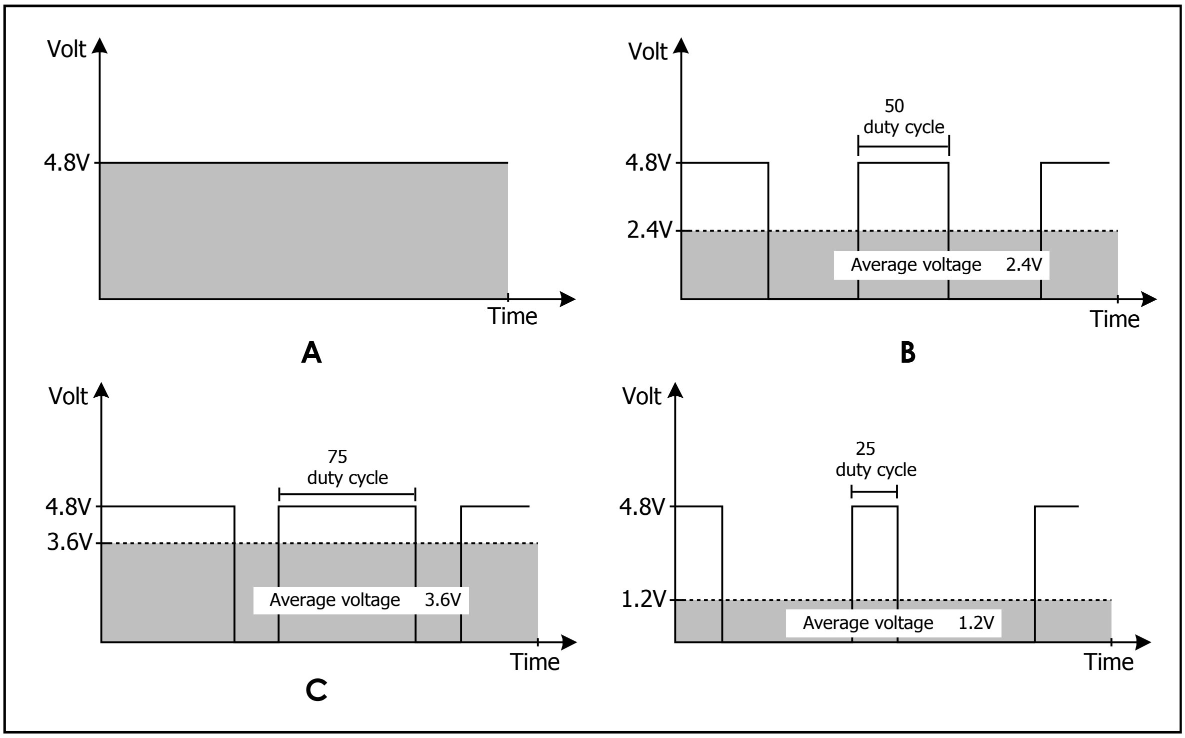
Acti it 5

Speed control of Robo-PICA

Robo-PICA can control the speed movement b send the signal to the enable pin (E) of motor driver IC, L2 3 . Refer the figure 4-1 (in this chapter), E pin of L2 3 is connected to RC2 CCP1 and RC1 CCP2 port pins of PIC16 877. Both port pins are P output port. Buiders can write the program to control the P output signal for adjustment motor speed.

P operation

ormal driving motor techni ue is appl the voltage to motor directl . The motor works in full speed. Sometime this speed faster. Then the simple method to control motor speed is control the voltage applied to motor. The populate techni ue is P (pulse-width modulation). This techni ue will control the width of the positive pulse. The voltage is applied to motor as average value. Ratio of positive pulse width and totall pulse width is called ut c cle. Its unit is percentage ( )



igure A5-1 Shows average voltage output of PWM

A ull volatge apply. B 50 duty cycle PWM

C 75 duty cycle PWM D 25 duty cycle PWM

```
'
'   Name    : P0501.BAS
'   Notes   : Speed Control
'
@ DEVICE PIC16F877,HS_OSC            used PIC16F877
DEFINE  OSC 20                       Oscillator = 20 MHz
OUTPUT  PORTC.1                      RC1 = OUTPUT BEFORE USE CH 2

LOOP:
     HP M        1 ,(100 255)/100,20000    Alway ON
     HP M        2 ,(100 255)/100,20000    Alway ON
       Motor Forward
     FREQOUT     PORTA.4,200,2000                Beep
     LO          PORTD.0 :   HIGH  PORTD.1       ..... Motor A
     HIGH        PORTD.2 :   LO    PORTD.3       ..... Motor B
     PAUSE       5000                            Delay 5 Second

     FREQOUT     PORTA.4,200,2000                Beep
     HP M        1 ,(90 255)/100,20000           20kHz, Duty Cycle 90%
     HP M        2 ,(90 255)/100,20000           20kHz, Duty Cycle 90%
     PAUSE       5000                            Delay 5 Second

     FREQOUT     PORTA.4,200,2000                Beep
     HP M        1 ,(80 255)/100,20000           20kHz , Duty Cycle 80%
     HP M        2 ,(80 255)/100,20000           20kHz , Duty Cycle 80%

     PAUSE       5000                            Delay 5 Second
     GOTO        LOOP
```

Listing 5-1 The speed control program for Robo-PICA by using PWM technique

In generation P singnal of PICBASIC PR compiler use HP command. The P signal will out from RC2 and RC1pin. See the listing P5-1 for example program.

A5.1 rite the BASIC code following the listing P4-1. Compile and download program to Robo-PICA.

A5.2 Select to R mode. bserve the operation.

The Robo-PICA robot will move fastest in 5 seconds and decreas speed After that the robot will back to increase the speed again The robot will move this ro tine all times

he suita le P duty cycle alue for dri ing the ro ot is more than 7
If select the less val e the robot has not tor e more eno gh for t rning or rotation

Program operation

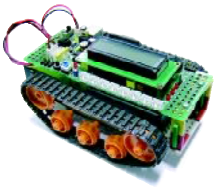
1. Set bit 1 of P RTC (RC1) as output. or RC2 will be set as output automatic in P operation.

2. Set P fre uenc as 20kHz, ut c cle 100 .

3. rive a beep and control the robot move forward 5 seconds.

4. rive a beep again. Change the P dut c cle to 0 for decreasing speed 5 seconds.

5. rive a beep and change P dut c cle to 80 for decreasing speed more. rive motor 5 seconds. Then back to step 3 again.



Acti it 6
Programmable mo ement control

ovement control in each robot depends on driving s stem. The Robo-PICA robot use C motor gearbox. It is open loop control. In controlling must select the suitable time value for each movement. However this time value must think about the motor initialize before motor start and inertialize after motor stop operation.

A6.1 rite the BASIC code following the listing P6-1. Compile and download program to Robo-PICA.

A6.2 Select to R mode. bserve the operation.

The Robo-PICA robot move forward 2 seconds t rn left 90 degree move forward 2 seconds t rn right 90 degree move backward 2 seconds t rn right 90 degree and loop to first step again

In real orld, each ro ot cannot rotate accurate at any alue In acti ity program define the time alue for turning degree at 6 millisecond Some ro ot can turn and some ro ot cannot ecause the effect from some factor such as attery po er le el and floor friction hen each uilder must change the suita le time alue for their ro ot

Beca se the robot se battery to power so rce In d ring the battery level is f ll power and not f ll the speed of movement is not e al It ca se the distance from movement may be not e al It is limitation of all robot that se open loop movement control

```
'
'   Name    : P0601.BAS
'   Notes   : Control Movement
'
@ DEVICE PIC16F877,HS_OSC            used PIC16F877

DEFINE  OSC  20                      Oscilator = 20 MHz
HIGH PORTC.2                         Enable Motor A
HIGH PORTC.1                         Enable Motor B
TRISD = %11110000                    RD0-RD3 = OUTPUT
Loop:
     GOSUB FOR ARD : PAUSE 2000      Forward 2 second
     GOSUB S_LEFT90                  Turn Left 90 deg.
     GOSUB FOR ARD : PAUSE 2000      Forward 2 second
     GOSUB S_RIGHT90                 Turn Right 90 deg.
     GOSUB BAC  ARD : PAUSE 2000     Backward 2 second
     GOSUB S_RIGHT90                 Turn Right 90 deg.
     GOTO  Loop                      Do again

FOR ARD:   PORTD = %00000110 : RETURN            Forward routine
BAC  ARD:  PORTD = %00001001 : RETURN            Backward routine
S_LEFT90:  PORTD = %00000101 : PAUSE 600 : RETURN
                                                 Turn left 90 deg. routine
S_RIGHT90: PORTD = %00001010 : PAUSE 600 : RETURN
                                                 Turn right 90 deg. routine
```

Listing 6-1 Time control robot movement experimental program for Robo-PICA

Chapter 5
Line following mission

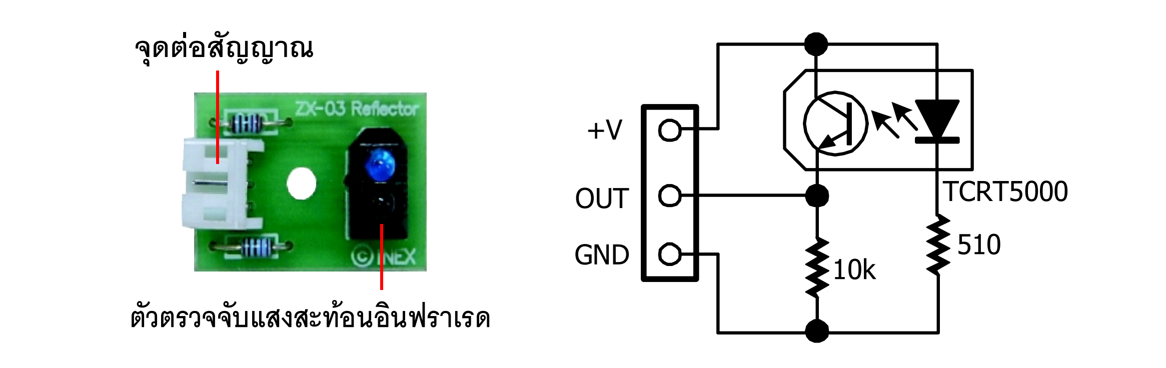
The popular activity of the educational robot is Line following or Line tracking. Purpose of this activity is learn about how to interface analog sensor. In Robo-PICA robot kit prepares a pair of Infrared reflector sensor for this activity. Add senses to the Robo-PICA so that it can detect and move following the line, by using the IR Reflector Sensor. Two IR Reflector Sensors will be installed at the bottom of the Robo-PICA so that it can detect both white and black lines.

5.1 ZX-03 Infrared Reflector

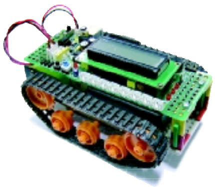
The heart of this sensor is TCRT5000 reflective object sensor. It is designed for close proximity infrared (IR) detection. There's an infrared diode behind its transparent blue window and an infrared transistor behind its black window. When the infrared emitted by the diode reflects off a surface and returns to the black window, it strikes the infrared transistor's base, causing it to conduct current. The more infrared incident on the transistor's base, the more current it conducts.

When used as an analog sensor, the ZX-03 can detect shades of gray on paper and distances over a short range if the light in the room remains constant.

The suitable distance from sensor to line or floor is during 3 to 8 mm. The output voltage is during 0.1 to 4.8V and digital value from10-bit A/D converter is 20 to 1,000. Thus, ZX-03 will suitable to apply to line tracking sensor.



รูปที่ 5-1 แสดงรูปร่างและวงจรของแผงวงจรตรวจจับแสงสะท้อนอินฟราเรด



Activity 7
Reading the Analog signal

The Robo-PICA's brain is PIC16F877 microconttroller. It contains 8-channel 10-bit anlog to digital converter module (ADC). All analog input ports can config to digital input and output. They include RA0-RA3, RA5, RE0-RE2. The important register of this module is **ADCON1 register.** The deail of this register see below

bit	7	6	5	4	3	2	1	0
	ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0
	R/W-0				R/W-0	R/W-0	R/W-0	R/W-0

R/W-0 : readable/writeable. RESET as logic"0"

If writitng program with PICBASIC PRO compiler, set the value for ADCON0 by themself not nescessary. Only use DEFINE ADC command to set the port function at 4-bit lower of ADCON1 (PCRFG3-PCFG0). See the setting at table 5-1

Bit 7 of ADCON1 register is ADFM bit. This bit use for selection the result data format. As "0", the 10-bit result will be Left justified. 6 Least Significant bits of the result are read as '0'. As "1" the 10-bit result will be Right justified. 6 Most Significant bits of the result are read as '0'.

10-bit Result

b	b	b	b	b	b	b	b	b	b	0	0	0	0	0	0	ADFM = "0" Left justified result
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

10-bit Result

0	0	0	0	0	0	b	b	b	b	b	b	b	b	b	b	ADFM = "1" Right justified result
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

The result data format from ADFM bit selection of ADCON1 register

PCFG3-PCFG0	AN7 (RE2)	AN6 (RE1)	AN5 (RE0)	AN4 (RA5)	AN3 (RA3)	AN2 (RA2)	AN1 (RA1)	AN0 (RA0)	Vref+	Vref -	Analog Ch. : Vref
0000	A	A	A	A	A	A	A	A	VDD	VSS	8:1
0001	A	A	A	A	Vref+	A	A	A	AN3	VSS	7:1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5:0
0011	D	D	D	A	Vref+	A	A	A	AN3	VSS	4:1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3:0
0101	D	D	D	D	Vref+	D	A	A	AN3	VSS	2:1
011X	D	D	D	D	D	D	D	D	-	-	0:0
1000	A	A	A	A	Vref+	Vref-	A	A	AN3	AN2	6:2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6:0
1010	D	D	A	A	Vref+	A	A	A	AN3	VSS	5:1
1011	D	D	A	A	Vref+	Vref-	A	A	AN3	AN2	4:2
1100	D	D	D	A	Vref+	Vref-	A	A	AN3	AN2	3:2
1101	D	D	D	D	Vref+	Vref-	A	A	AN3	AN2	2:2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1:0
1111	D	D	D	D	Vref+	Vref-	D	A	AN3	AN2	1:2

A : Analog input, **D** : Digtial Input/Output,
Vref+ : Positive reference voltage input, **Vref-** : Negative reference voltage input,
VDD : Reference voltage as Supply voltage, **VSS** : Reference voltage as Ground,
AN3 : Analog input 3 as Vref+, **AN2** : Analog input 2 as Vref-

Table A5-1 Selection bit PCFG3-PCFG0 of ADCON1 register summary

Interface A/D converter module in PIC16F877 with PICBASIC PRO compiler

PICBASIC PRO compiler provides ADCIN command to interface A/D Converter module. Its syntax is

```
ADCIN Channel,Var
```

as : `Channel` is Analog input. Select 0 to 7

`Var` is Variable contains the result from conversion. It is 16-bit variable.

In addition, user can use `DEFINE` command for adjust the operation of A/D Converter as :

`DEFINE ADC_BITS` - **Select bit number reading.** Normally select 8, 10 or 12 bits depend on microcontroller. For PIC16F877 can select 8 or 10 bits. The resolution will change following bit number, 8-bit resolution is 256 numbers, 10-bit resolution is 1,024 numbers and 12-bit resolution is 4,096 numbers.

`DEFINE ADC_CLOCK` - **Select A/D conver clock source.** Select 0 to 3 as :

"0" : FOSC/2

"1" : FOSC/8

"2" : FOSC/32

"3" : FRC (Internal RC clock)

`DEFINE ADC_SAMPLEUS 50` - **Select delay for conversion time.** Unit is microsecond. Insert `PAUSEUS` into delay routine during setting Channel value and start conversion.

How to read sensor value ?

In making Robo-PICA from activity 2 both of ZX-03 Infrared reflector are installed at bottom of the robot body and connect the sensor's cable to RA0 and RA1 port pin. Next activity is writing the program to read sensor's value to display at LCD module of Robo-PICA.

A7.1 Connect LCD module on the RBX-877 board of Robo-PICA robot.

A7.2 Write the BASIC code following the listing P7-1. Connect the download cable to the robot. Turn-on power. Select MODE switch to program mode. Click button or press F10 key for compile and download program to Robo-PICA.

A7.3 Select to RUN mode (green LED at RUN label lights). Observe the operation at LCD module.

LCD module shows message `SENSOR0 = xxx` *(*`xxx`* as 0 to 1023)*

A7.4 Turn-off POWER switch and unplug the download cable.

A7.5 Place Robo-PICA on the white surface. The position of ZX-03 that connect to RA0 port must above the white surface. Turn-on POWER switch. See the result at LCD screen and record it.

A7.6 Place Robo-PICA on the black surface. The position of ZX-03 that connect to RA0 port must above the black surface. Turn-on POWER switch. See the result at LCD screen and record it.

At the black surface or line, the digital value that get from Infrared reflector sensor (ZX-03) will be lower 200. But for the white surface or line value will be upper 600.

```
'*****************************************************************
'*   Name     : P0701.BAS                                        *
'*   Notes    : Test Direction of Motor                          *
'*****************************************************************
@ DEVICE  PIC16F877,HS_OSC     ' used PIC16F877
SENSOR0     VAR         WORD
DEFINE      OSC         20     ' Oscilator = 20 MHz
DEFINE      ADC_BITS    10     ' Set number of bits in result
DEFINE      ADC_CLOCK   3      ' Set clock source (Internal RC = 3)
DEFINE      ADC_SAMPLEUS  50   ' Set sampling time in microseconds
DEFINE      LCD_DREG    PORTD  ' Set LCD Data port
DEFINE      LCD_DBIT    4      ' Set starting Data bit (0 or 4) if 4-bit bus
DEFINE      LCD_RSREG   PORTC  ' Set LCD Register Select port
DEFINE      LCD_RSBIT   0      ' Set LCD Register Select bit
DEFINE      LCD_EREG    PORTC  ' Set LCD Enable port
DEFINE      LCD_EBIT    5      ' Set LCD Enable bit
DEFINE      LCD_BITS    4      ' Set LCD bus size (4 or 8 bits)
DEFINE      LCD_LINES   2      ' Set number of lines on LCD
DEFINE      LCD_COMMANDUS     2000  ' Set command delay time in us
DEFINE      LCD_DATAUS  50     ' Set data delay time in us

            ADCON1 = %10000000       ' All PORTA & PORTE is Analog

MAIN:ADCIN 0, SENSOR0                ' Read channel 2
            LCDOUT $FE,$01,"SENSOR0 = ", DEC SENSOR0
            PAUSE  500
            GOTO   MAIN
```

Listing P7-1 The program reading ZX-03 value from RA0 channel to display on LCD module

A7.7 Edit the listing P7-1 to read data from sensor at RA1 port from

```
ADCIN 0 , SENSOR0
```

to

```
ADCIN 1 , SENSOR0
```

Connect the download cable to the robot. Turn-on power. Select MODE switch to program mode. Click button or press F10 key for compile and download program to Robo-PICA again.

A7.8 Select to RUN mode. See the result at LCD module and record. Consider the result same or different from value of RA0 (in step A7.3).

If different value between white and black surface reflection is less. Builder must adjust the distance from sensor to surface decreasing. If the sensor far from surface more, value will near zero.

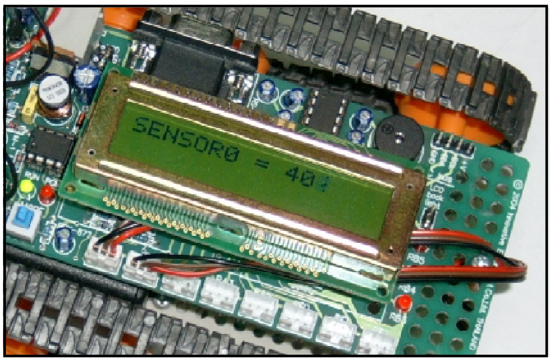


Figure A7-1 Shows the result from reading the ZX-03 Infrared reflector sensor operation on LCD module of Robo-PICA

From testing can conclude the result as :

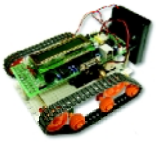
The white surface reading is during 400 to 900

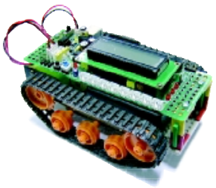
The black surface reading is during 0 to 150

Thus, the reference value for making decision is during 150 to 400. Builder can select the suitable value and make decision as :

If the sensor reading value more than the reference value, sensor detect ***"WHITE surface"***

If the sensor reading value less than the reference value, sensor detect ***"BLACK surface"***

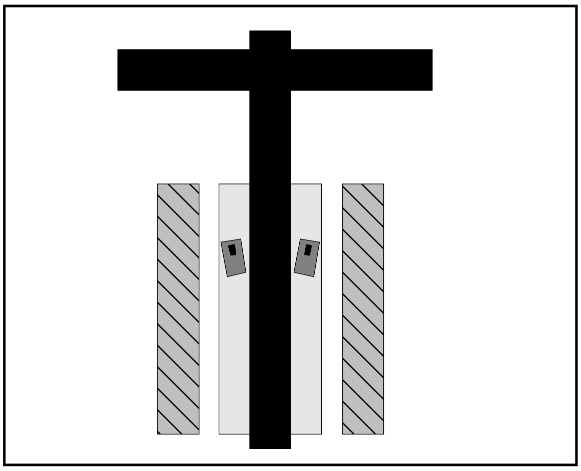




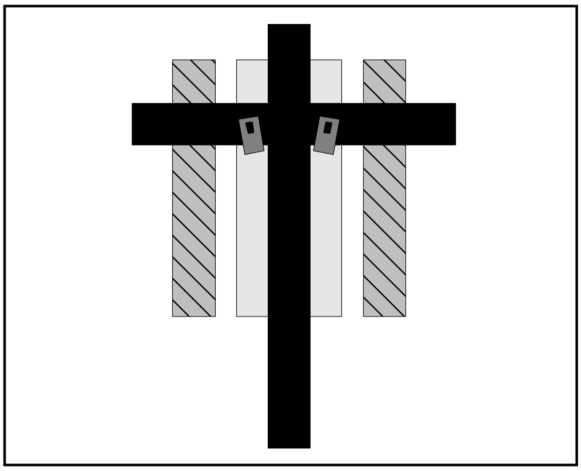
Activity 8

Robo-PICA moves follow the black line

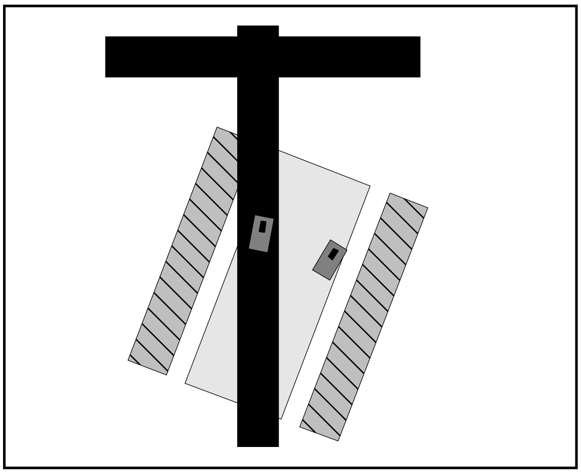
From installation ZX-03 Infrared reflector in Robo-PICA can set the line following behavior to 4 scenario. See the figure below.



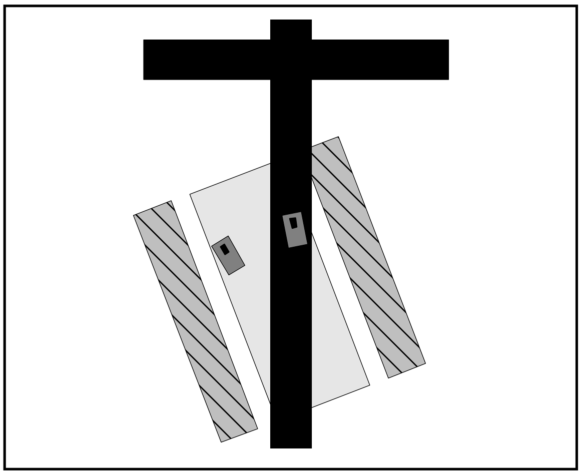
1. Both sensor above the white surface This scenario can interpret the robot move bestride the line. The robot must be controlled so that it moves forward and delays briefly.



2. Both sensor above the black surface This scenario can interpret the robot stay on the accross line. The robot has many options to choose from whether it is to move forward, turn left or turn right.



3. The left sensor is above the black floor. This scenario can interpret the robot moves rightwards away of the line. The robot must be controlled so that it turns left slowly and delays briefly.



4. The right sensor is above the black floor. This scenario can interpret the robot moves leftwards away of the line. The robot must be controlled so that it turns right slowly and delays briefly.

How to find the reference value ?

If the sensor reading value less than the reference value, can interpret the deteced surface as **“Black”**

If the sensor reading value more than the reference value, can interpret the detected surface as **“White”**

The reference value can calulate from :

(The sensor’s minimum reading value from white surface + The sensor’s maximum reading value from black surface) / 2

Example

If The sensor’s minimum reading value from white surface is equal 300 and The sensor’s maximum reading value from black surface is equal 100

The reference value will be equal (300+100) / 2 = **200**

From the black line following behavior can apply to make the program to control the Robo-PICA. See the listing P8-1. The sequence of program operations are

1. Set the default value to interface A/D Converter module in PIC16F877 microcontroller

2. Enable both DC motor driver circuits.

3. Get data from both sensors that connected at RA0 and RA1 port pins. Load data to SENSOR0 (store data from RA0) and SENSOR1 (store data from RA1) variable

4. Compare readig data with the reference data. *If SENSOR0 and SENSOR1 data are more than the reference, control the robot to move forward.*

5. *If SENSOR0 and SENSOR1 data are less than the reference, control the robot to move forward.* This condition is the robot meets an intersection and decide to move forward continue.

6. *If SENSOR0 data is less than the reference only, the robot turns left.*

7. *If SENSOR1 data is less than the reference only, the robot turns right.*

```
'****************************************************************
'*   Name     : P0801.BAS                                       *
'*   Notes    : Track Line                                      *
'****************************************************************
@ DEVICE  PIC16F877,HS_OSC    ; used PIC16F877
SENSOR0    VAR   WORD
SENSOR1    VAR   WORD

DEFINE     OSC               20          ; Oscilator = 20 MHz
DEFINE     ADC_BITS          10          ; Set number of bits in result
DEFINE     ADC_CLOCK         3           ; Set clock source (Internal RC = 3)
DEFINE     ADC_SAMPLEUS      50          ; Set sampling time in microseconds

           HIGH  PORTC.2                 ; Enable Motor A
           HIGH  PORTC.1                 ; Enable Motor B
           TRISD = %11110000             ; RD0-RD3 = OUTPUT
           ADCON1 = %10000000            ; All PORTA & PORTE is Analog

MAIN:ADCIN 0,SENSOR0                     ; Read channel 0
           ADCIN 1,SENSOR1               ; Read Channel 1

           IF (SENSOR0 > 200) AND (SENSOR1 > 200) THEN Forward
           IF (SENSOR0 < 200) AND (SENSOR1 < 200) THEN Forward
           IF (SENSOR0 < 200) THEN S_Left
           IF (SENSOR1 < 200) THEN S_Right
           GOTO MAIN

FORWARD:   PORTD = %00000110 : GOTO MAIN       ; Forward Routine
BACKWARD:  PORTD = %00001001 : GOTO MAIN       ; Backward Routine
S_LEFT:    PORTD = %00000101 : GOTO MAIN       ; Turn Left Routine
S_RIGHT:   PORTD = %00001010 : GOTO MAIN       ; Turn Right Routine
```

Listing P8-1 Black line following demonstration program of Robo-PICA

A8.1 Write the BASIC code following the listing P8-1. Connect the download cable to the robot. Turn-on power. Select MODE switch to program mode. Click button or press F10 key for compile and download program to Robo-PICA.

A8.2 Turn-off POWER switch and unplug the download cable.

A8.3 Place Robo-PICA cross the black line in the demonstration paper filed (bundled in Robo-PICA robot kit).

A8.4 Turn-on POWER switch and select to RUN mode. Observe the robot operation.

The Robo-PICA will move following the black line and not detect the across line or intersection line.

Summary

The heart of line following robot operation are 3 factors. First, the sensor's performance. Second, sensor's installation and Third, the control software. Sometimes the robot need more sensors to detect the complex line such as 3 cross line, intersection line, etc. See the figure A8-1

About installation Infrared reflector to detect the line, the distance between both sensors is important. The suitable space is improve the detection. If the sensor is near the line more, the robot will detect the line many times. It cause the robot movement swing from left to right similar snake movement. See the figure A8-2. It means the speed of movement reduce.

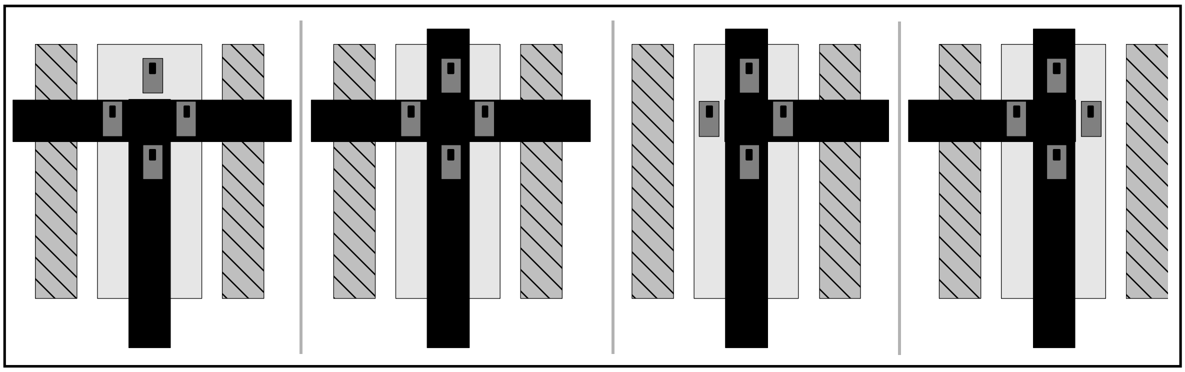
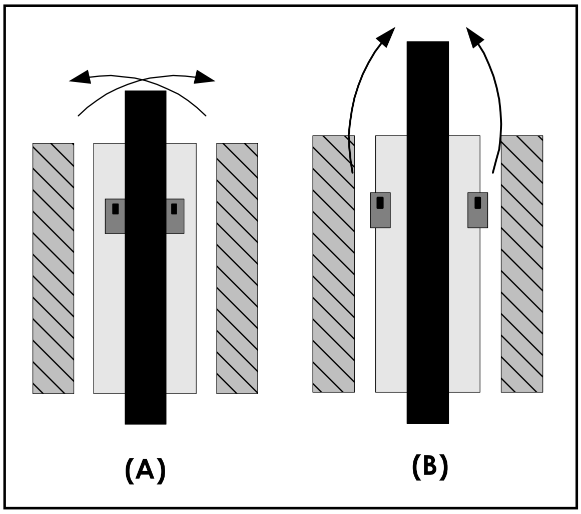


Figure A8-1 Add more line tracking sensors to detect the complex line.



Figue A8-2 The result of installation line detector.

(A) In case install the sensor near over. It cause the robot's movement swing.

(B) Installation both sensors far from them and line. If good enough, it help the robot move and following the line better.

Chapter 6

Robot with Remote control

Another feature of automatic robots is that it can receive commands from a far distance b using infrared light. This is similar to a remote-controlled robot except that the commands received are through serial communication. Robo-PICA provides an Infrared Remote control, called ER-4. ER-4 remote control will modulate serial data with infrared light. Robo-PICA must install a 38kHz Infrared receiver module for receiving.

6.1 X- 5 Infrared Recei er module

In transmit the data modulated with infrared light for long distance about 5 to 10 metres similar T remote control. The carrier fre uenc is 38kHz.Thus, at receiver must demodulate 38kHz carrier fre uenc . After that transfer serial data to microcontroller.

If the sensor does not detect the 38kHz fre uenc with the infrared light, the output will be logic 1 . therwise, if it detects the 38kHz fre uenc , the output logic is 0 .

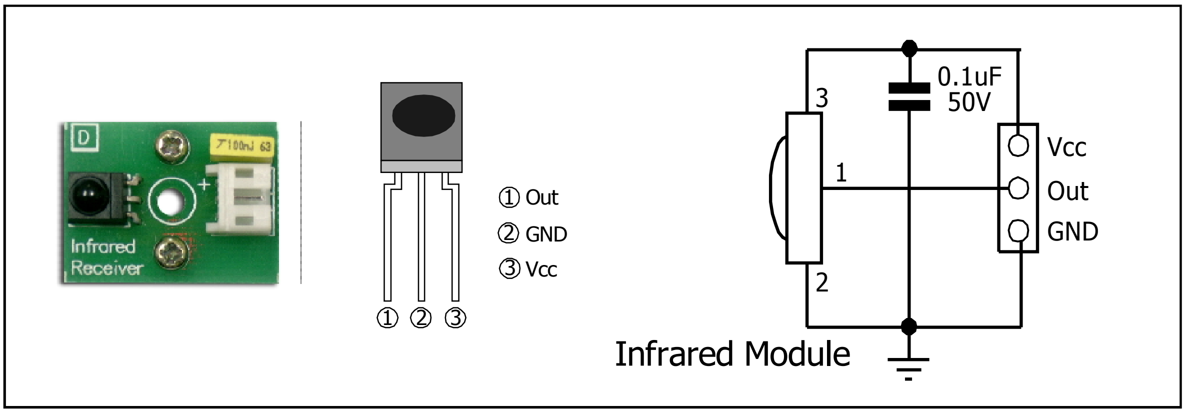


Figure 6-1 Shows the photo of 38kHz Infrared Receiver module pin assignment and schematic diagram

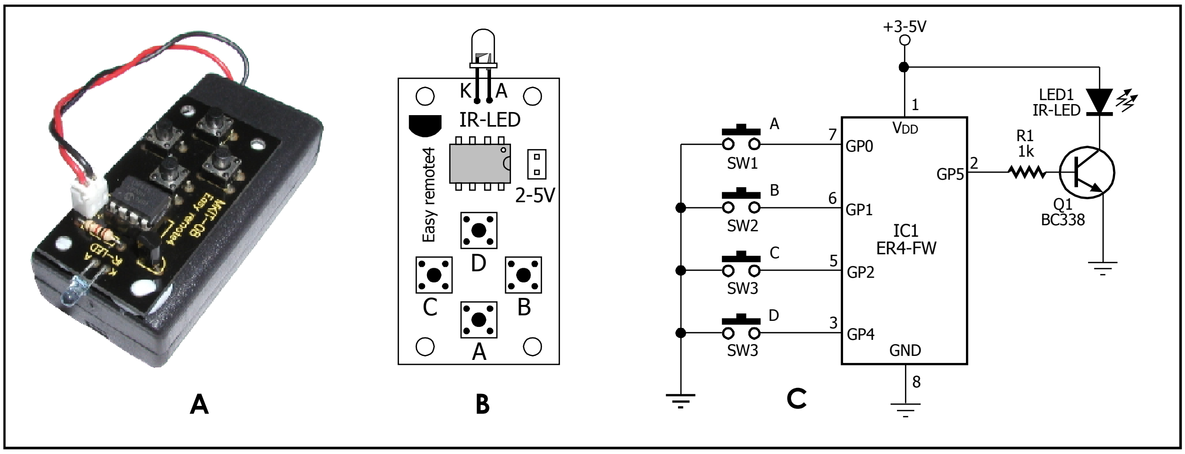


Figure 6-2 Shows the photo board layout and Schematic of ER-4 Easy remote control

6.2 ER-4 Infrared Remote control

- perational distance is 4 to 8 meters in open space.
- The 4-channel switch operates in an on off mode
- ses low power Automaticall   resumes power-save mode once data is sent
- ses onl   2.4-3.0   from two AA batteries - both regular and rechargeable.
- Transmits serial data using the RS-232 standard with 1200 bps baud rate and 8   1 data format (8 data bit, no parit , 1 stop bit)

6.2.1 Format of data sent b   Eas   Remote4

To make it easier for the receiver to read the switch value from the remote control, the ER-4 transmit serial data according to the RS-232 standard, with a baud rate of 1,200 bps and 8   1 format. Characters are transmitted according to what switch is pressed on the remote. The switch positions are displa ed in   igure 6-2

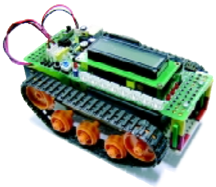
Press switch A, the large cap A , followed b   small cap A (a) is sent.

Press switch B, the large cap B, followed b   small cap B (b) is sent

Press switch C, the large cap C, followed b   small cap C (c) is sent

Press switch , the large cap , followed b small cap   (d) is sent.

The reason that we have to alternate large cap and small cap letters is so that the receiver can differentiate if a user presses continuousl   or if the user represses. If a user represses, the large cap character will be sent the first time. If the user represses the same button again, the small cap character will be sent the second time If the user presses continuall , the last character will be sent repeatedl .



Acti it

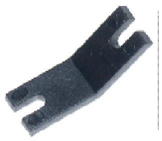
Install 38k Infrared recei er to Robo-PICA

Part list

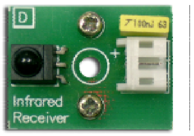
mm la tic acer

mm Screw

mm ut

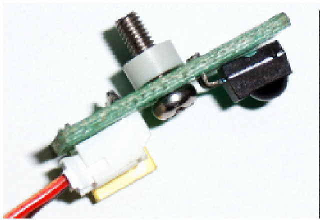


tu e oiner

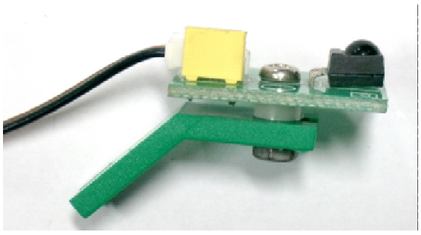


n rared ecei er module

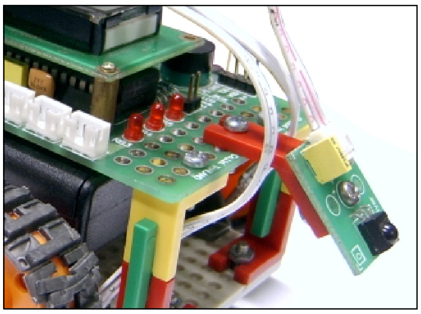
Construction



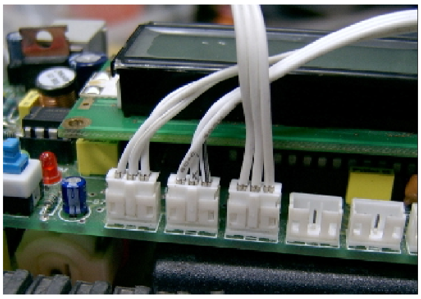
A .1 se a 3 x 10 mm screw and place it into the Infrared Receiver odule, followed b the 3 mm



A .2 Then place the obtuse joiner at the back of the module and used a 3 mm nut to screw it all in together tightl .

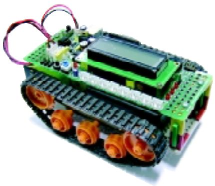


A .3 Attach the Infrared Receiver module from step A .2 in front of the robot b 3x10 mm. screw and nut.



A .4 Plug sensor cable to RA2 port. In programming, define the P RTA.2 to digital input for reading data from 38kHz Infrared receiver module.

From here Robo-PICA can communicate the external de ice ia infrared in serial data.



Acti it 1

Reading the ER-4 Remote control data

In listing P10-1 is BASIC program for reading data from ER-4 remote control and shows on LC screen. Addition use this data to compare the reference data for driving sound to piezo speaker b RE T command in PICBASIC PR compiler.

A10.1 rite the BASIC code following the listing P10-1. Compile and download program to Robo-PICA.

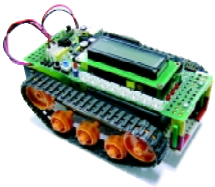
```
'
'   Name     : P1001.BAS
'   Notes    : Use SERIN2 Receive Data From Remote Control
'
@ DEVICE PIC16F877,HS_OSC          ' used PIC16F877
 EY         VAR         BYTE
DEFINE      OSC         20         ' Oscilator = 20 MHz
DEFINE      LCD_DREG    PORTD      ' Set LCD Data port
DEFINE      LCD_DBIT    4          ' Set starting Data bit (0 or 4)
DEFINE      LCD_RSREG   PORTC      ' Set LCD Register Select port
DEFINE      LCD_RSBIT   0          ' Set LCD Register Select bit
DEFINE      LCD_EREG    PORTC      ' Set LCD Enable port
DEFINE      LCD_EBIT    5          ' Set LCD Enable bit
DEFINE      LCD_BITS    4          ' Set LCD bus size (4 or 8 bits)
DEFINE      LCD_LINES   2          ' Set number of lines on LCD
DEFINE      LCD_COMMANDUS 2000     ' Set command delay time in us
DEFINE      LCD_DATAUS  50         ' Set data delay time in us

ADCON1 = $07                       ' ALL PORTA & PORTE = Digital I/O
MAIN:
     SERIN2 PORTA.2,813,2000,MAIN,[ EY]   ' Receive Data From IR Module

     LCDOUT  $FE,$01,"  EY = ",  EY       ' Show Data On LCD Module

     IF ( EY = "A") OR ( EY = "a") THEN   ' Check  ey = "A" or "a" ?
           FREQOUT PORTA.4,100,2000       ' 2kHz Beep
           ENDIF
     IF ( EY = "B") OR ( EY = "b") THEN   ' Check  ey = "B" or "b" ?
           FREQOUT PORTA.4,100,2200       ' 2.2kHz Beep
           ENDIF
     IF ( EY = "C") OR ( EY = "c") THEN   ' Check  ey = "C" or "c" ?
           FREQOUT PORTA.4,100,2400       ' 2.4kHz Beep
           ENDIF
     IF ( EY = "D") OR ( EY = "d") THEN   ' Check  ey = "D" or "d" ?
           FREQOUT PORTA.4,100,2600       ' 2.6kHz Beep
           ENDIF

     GOTO MAIN
```

Listing 1 -1 Reading data from ER-4 remote control demonstration program

A10.2 Select to R mode.

A10.3 Press switch on ER-4 remote control for sending signal to 38kHz Infrared Receiver module at Robo-PICA. Thev direction must direct. bserve the operation at LC module and piezo spekaer.

C will show letter A B C or a b c d following press switch on ER-4 Remote control and listen the diffrent beep so nd fre ency



Acti it 11

IR control Robo-PICA's mo ement

This activit shows how to control Robo-PICA s movement via ER-4 Infrared remote control. The switch or button on ER-4 remote control will function move forward-backward-left-right.

A11.1 rite the BASIC code following the listing P11-1. Compile and download program to Robo-PICA.

A11.2 Select to R mode.

irstly Robo-PICA will stay ntil pressing b tton on ER-4 Remote control The direction of sending light m st straigth The comm nication will be complete

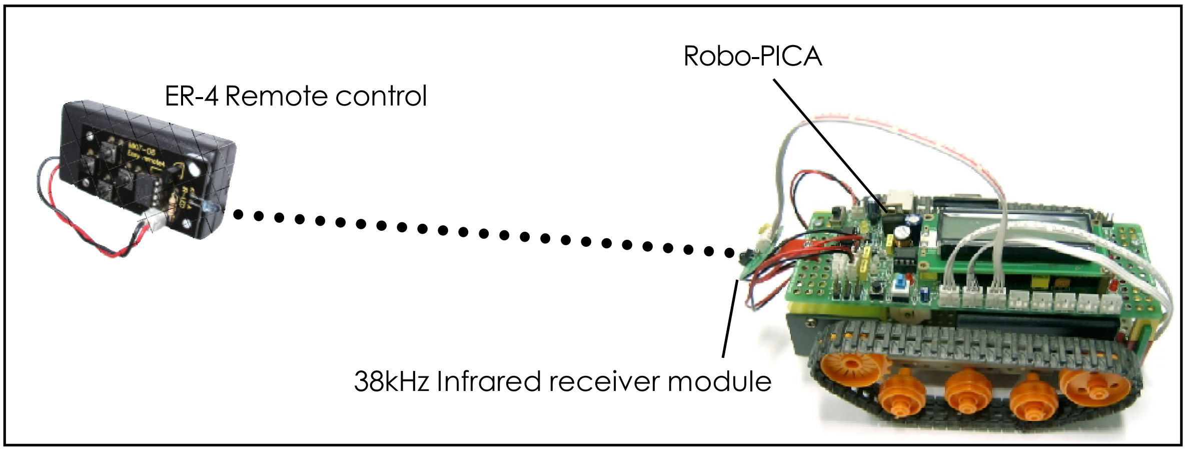


Figure A11-1 Shows controlling the Robo-PICA with ER-4 Infrared remote control.

```
'
'   Name    : P1101.BAS
'   Notes   : Control ROBO-PICA  ith Remote Control
'
@ DEVICE PIC16F877,HS_OSC           ' used PIC16F877
 EY  VAR   BYTE
DEFINE OSC 20                       ' Oscilator = 20 MHz
ADCON1 = $07                        ' ALL PORTA & PORTE = Digital I/O
HIGH PORTC.2                        ' Enable Motor A
HIGH PORTC.1                        ' Enable Motor B
TRISD = %11110000                   ' RD0-RD3 = OUTPUT
MAIN:
     SERIN2 PORTA.2,813,100,FREE,[ EY]    '  ait Recieve Motor
                                          ' If No Data Stop Motor
     IF ( EY = "A") OR ( EY = "a") THEN Backward      ' Pess A or a action
     IF ( EY = "B") OR ( EY = "b") THEN S_Right       ' Press B or b action
     IF ( EY = "C") OR ( EY = "c") THEN S_Left        ' Press C or c action
     IF ( EY = "D") OR ( EY = "d") THEN Forward       ' Press D or d action
     GOTO MAIN

FREE:       PORTD = %00000000 : GOTO Main       ' Stop motor routine
FOR ARD:    PORTD = %00000110 : goto Main       ' Forward routine
BAC  ARD:   PORTD = %00001001 : Goto Main       ' Backward routine
S_LEFT:     PORTD = %00000101 : GOTO Main       ' Turn left routine
S_RIGHT:    PORTD = %00001010 : GOTO Main       ' Turn right routine
```

โปรแกรมที่ P11-1 โปรแกรมควบคุมหุ่นยนต์ Robo-PICA ให้เคลื่อนที่ด้วยข้อมูลอนุกรมที่ส่งมาจากรีโมตคอนโทรล ER-4

Program description

Start with define port A and E to digtial i o port b A C 1 07 command. Enable C motor driver circuits to drive motor A and B in full speed. ext, read data from 38kHz Infrared receiver module at RA2 with SERI 2 command. The data stores in E variable. If not press an ke in 100 millisecond, program will jump to work at REE routine. It is stop motor routine. After reading the data from RA2, compare the switch s reference data.

If E or d then move the robot forward

If E C or C then rotate the robot right

If E B or b then rotate the robot left

If E A or a then move the robot backward

To define all port A as digital inp t o tp t port m st set val e to A C 1 register as 0x07

Chapter 7

Contactless object detection of Robo-PICA

The one popular application of intelligent educational robot is contactless object avoiding. Robo-PICA also supports the application. The new sensor will suggest in this chapter is ZX-08 Infrared objector.

7.1 X- 8 Infrared objector

This sensor has both the infrared light receiver and transmitter within itself. The ZX-08 sensor can detect obstacles at a maximum distance of 6 centimeters. The Rx signal cable must be connected to the digital input, while the Tx must be connected to an digital output on the RBX-877 board. nce the signal cables are connected, sending logic 1 to Tx will make the infrared LE on the ZX-08 module light up. If there is an obstacle blocking in front, the infrared light will send reflect that object back to the infrared receiver, causing a logic 0 to be sent to the Robo-PICA input port.

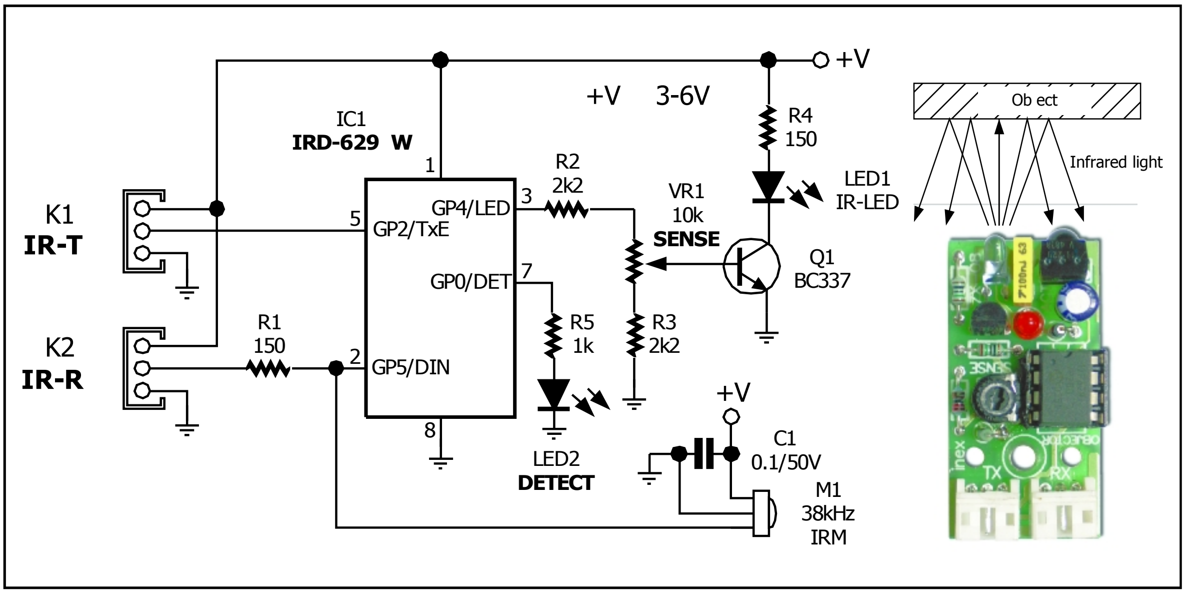
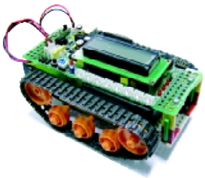


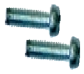
Figure 7-1 ZX-08 Infrared ob ector schematic



Acti it 12

Install X- 8 Infrared object detector

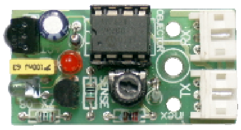
Part list



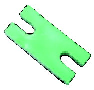
3x10 mm. Screw

3 mm. ut

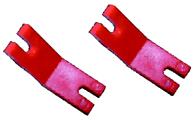
3 mm. Plastic spacer



ZX-08 Infrared objector



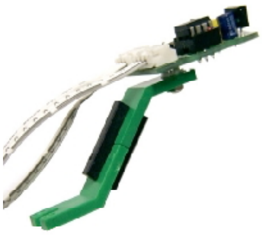
Straight joiner



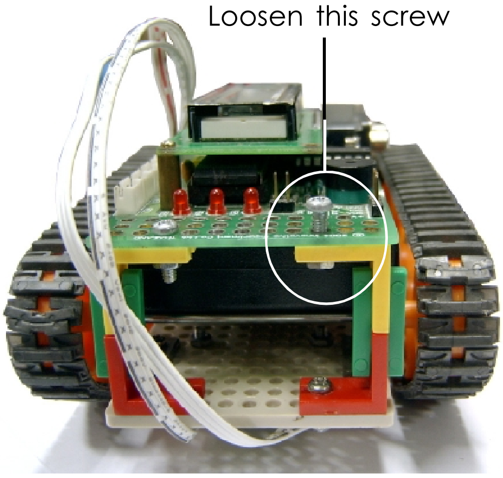
btuse joiner

Construction

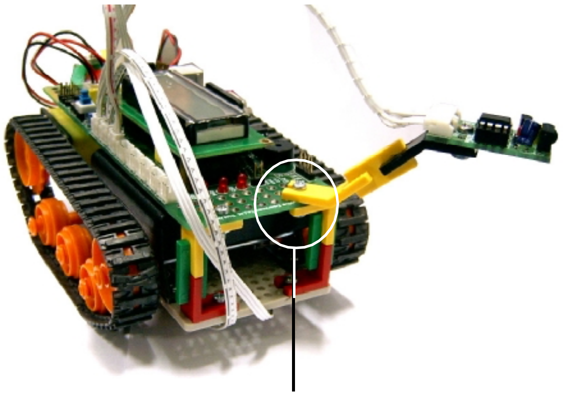
A12.1 ake the sensor structure before. Insert 3x10 mm. screw through ZX-08 board, 3 mm. spacer and a btuse joiner. Tighten b 3 mm. nut. Connect the other end of btuse joiner with Straight joiner following another btuse joiner finall . ake this ZX-08 structure 2 sets.



A12.2 Loosen a screw that attach the RBX-877 board with Right angle joiner and insert ZX-08 structure from step A12.1 to space between screw and RBX-877 board. Tighten this screw again for locking the structure. ake same method with another ZX-08 structure.



Loosen this screw



Insert the ZX-08 structure to space between screw and RBX-877 board

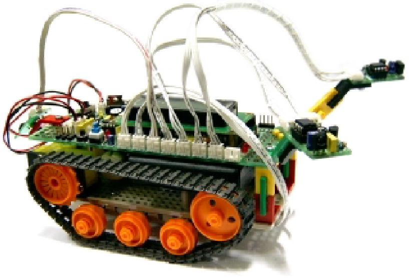
A12.3 Plug ZX-08 cable to RBX-877 as

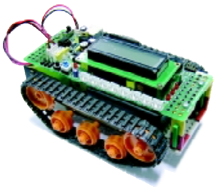
Rx of eft ZX-08 connect to RA3 port

Rx of Right ZX-08 connect to RA5

Tx of Left ZX-08 connect to RE0 port

Tx of Right ZX-08 connect to RE1 port





Acti it 13
esting Infrared objector

This activit demonstrates testing the operation of ZX-08 Infrared objector b get signal from sensor and displa the result on LC screen.

A13.1 rite the BASIC code following the listing P13-1. Compile and download program to Robo-PICA.

```
'
'   Name     : P1301.BAS
'   Notes    : Detect  all Show Status ON LCD
'
@ DEVICE PIC16F877,HS_OSC           ' Use PIC16F877

DEFINE     OSC          20          ' Oscilator = 20 MHz

DEFINE     LCD_DREG     PORTD       ' Set LCD data port
DEFINE     LCD_DBIT     4           ' Set starting Data bit
DEFINE     LCD_RSREG    PORTC       ' Set LCD register Select port
DEFINE     LCD_RSBIT    0           ' Set LCD register Select bit
DEFINE     LCD_EREG     PORTC       ' Set LCD enable port
DEFINE     LCD_EBIT     5           ' Set LCD enable bit
DEFINE     LCD_BITS     4           ' Set LCD bus size (4 or 8 bits)
DEFINE     LCD_LINES    2           ' Set number of lines on LCD
DEFINE     LCD_COMMANDUS 2000       ' Set command delay time in us
DEFINE     LCD_DATAUS   50          ' Set data delay time in us

ADCON1 = $07                        ' ALL PORTA & PORTE = Digital I/O
MAIN:HIGH PORTE.1                   ' Infrared LED ON
     HIGH PORTE.0
     IF (PORTA.3 = 0) THEN          ' Check IR MOdule detection
           LCDOUT  $FE,$01,"DETECT  ALL_L"
                                    ' If detect, show on LCD
     ENDIF
     IF (PORTA.5 = 0) THEN
           LCDOUT $FE,$01,"DETECT  ALL_R"
     ENDIF
     PAUSE 100                      ' Delay
     LCDOUT $FE,$01
     GOTO MAIN                      ' Check again
```

Listing 13-1 Infrared ob ector demonstration program of Robo-PICA

A13.2 Select to R mode.

A13.3 Place the object in front of the Robo-PICA. o not place far over 6 cm. If the red LE of ZX-08 turn-on. It means detection object read . If LEd not on, adjust the variable resistor on ZX-08 until LE can light. After ZX-08 works, LC will show message E E if work with Left ZX-08 and E E if work with Right ZX-08.

Abo t the detection range cannot specific It depends on material color and ambient light If object s color is dark the detection range will shorter In the other hands boject s color is bright s ch as white or yellow the dtection range will longer However ZX-08 can detect in range 3-6 cm

However installation of both ZX-08 structure in Robo-PICA must far enough for reducing the crosstalk from another side.

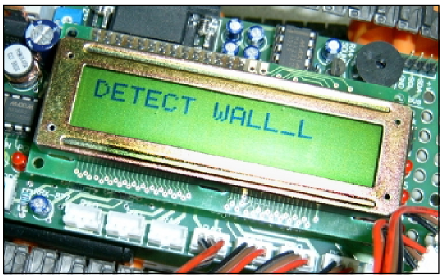
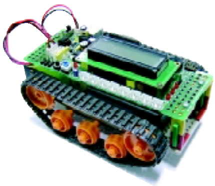


Figure A13-1 Shows testing of ZX-08 on LCD of Robo-PICA



Acti it 14

Contactless object detection robot

A14.1 rite the BASIC code following the listing P13-1. Compile and download program to Robo-PICA. Turn-off P ER switch and unplug download cable.

A14 2 Place the robot on the floor T rn-on P WER switch and select to R mode. See the robot operation.

Robo-PICA moves forward contin o s If it detect the object it will move backward and change to move at the opposite direction S ch as detect object at the left - the robot will change to move at the right diection If detect at the right - the robot will change to move at the left direction

```
'
'   Name     : P1401.BAS
'   Notes    : Detect  all
'
@ DEVICE PIC16F877,HS_OSC     ' used PIC16F877
DEFINE OSC 20                 ' Oscilator = 20 MHz

ADCON1 = $07                  ' ALL PORTA & PORTE = Digital I/O
HIGH PORTC.2                  ' Enable Motor A
HIGH PORTC.1                  ' Enable Motor B
TRISD = $F0                   ' Set low nibble PORTD for driving motor
MAIN:
     HIGH PORTE.1             ' Infrared LED on
     HIGH PORTE.0
     IF (PORTA.3 = 1) AND (PORTA.5 = 1) THEN FOR ARD '   Not   detect,   go
straight
     IF (PORTA.3 = 0) THEN LEFT                      ' Detect from left
     IF PORTA.5 = 0 THEN RIGHT                       ' Detect from right
     GOTO MAIN

FOR ARD:   PORTD = %00000110 :GOTO MAIN
LEFT:      PORTD = %00001001 : PAUSE 1000 : PORTD = %00000101 : PAUSE 800
           GOTO MAIN
RIGHT:     PORTD = %00001001 : PAUSE 1000 : PORTD = %00001010 : PAUSE 800
           GOTO MAIN
```

Listing 14-1 Contactless ob ect detecion robot demonstration program