

# SEED TECHNOLOGY INC (SEEEDUINO)

## Seeeduino Stalker v2.0

### Model: ARD104D2P

## Introduction

Seeeduino Stalker is a feature rich Arduino compatible Wireless Sensor Network node. It's modular structure and onboard peripherals makes it convenient to log time stamp sensor data on a periodic basis. The Seeeduino Stalker is a good candidate for all your tracking, monitoring and control projects. </br>

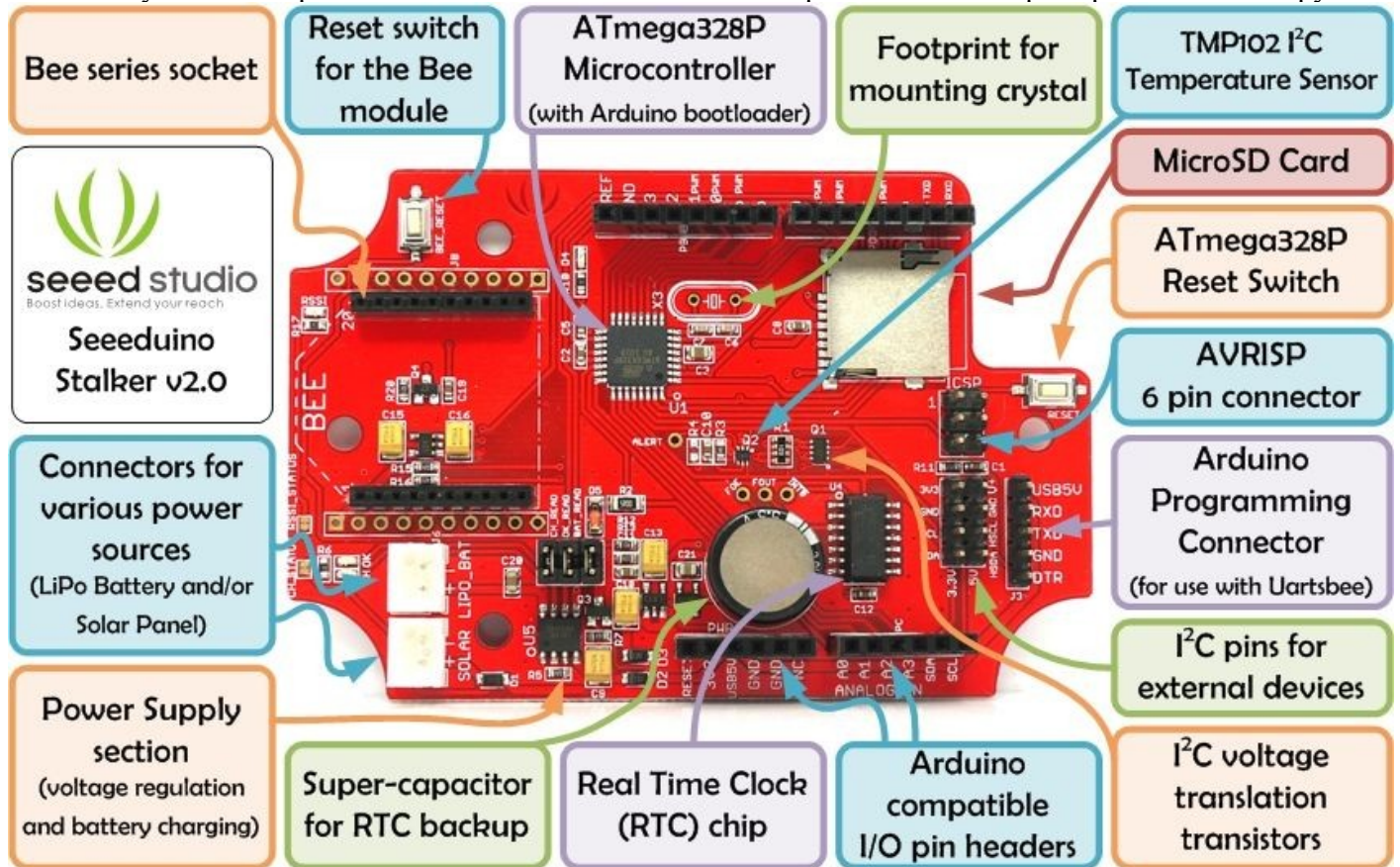


Seeeduino Stalker v2.0

</br>

## Features

- Compatible with Seeeduino.
- Mechanically compatible with Arduino(I/O ports are only at 3.3V)
- Onboard microcontroller: ATmega328P
- Onboard Real Time Clock chip with a supercap as a backup power source
- Serial interface with DTR for auto reset during programming when operating in standalone mode. (For programming, [UartSBee](#) must be bought separately)
- microSD card socket
- I2C Pin header (operation voltage is selectable: 5.0V or 3.3V)
- User defined switch and LED
- [Bee series socket](#) - 2\*10 pin 2.0mm pitch (which will mate with - one at a time - any of the wireless modules: [XBee](#), [BluetoothBee](#), [GPSBee](#) or [RFBee](#).)



Seeduino Stalker v2.0 - Various sections are marked

## Application Ideas



Seeduino Stalker v2.0 with the GPSBee inserted.

- Wireless Sensor Network (using [XBee](#) - bought separately)
- GPS Logging (using [GPSBee](#) - bought separately)
- Data Acquisition System capable of communicating with an App running on iPhone/Android Phone/Nokia Phone (using [BluetoothBee](#) - bought separately)
- RF Remote Control (using [RFBee](#) - bought separately)
- As a simple standalone Arduino compatible physical computing platform ([UartSBee](#) must be bought separately for programming)



**WiFiBee**

**WiFi MAC module (MRF24WB0MA based)**  
For adding WiFi to your Stalker. Connects to the microcontroller via SPI. TCP/IP stack needs to be added separately.

**UartSBee**

**USB-UART Bridge (FT232RL based)**  
For programming (using Arduino IDE) and serial communication with other Bee modules. Can be used with other Bee modules to connect them to PC directly.

**BluetoothBee**

**Bluetooth SPP Module (BlueCore4 based)**  
For adding a Bluetooth Serial Port to your Stalker. Connects to the microcontroller via UART.

**XBee / XBee ZB**

**Proprietary RF / ZigBee Module**  
For adding a radio interface meant for wireless sensor mesh networks to your Stalker. Connects to the microcontroller via UART. Accepts AT commands.

**GPSBee**

**Global Positioning System Module**  
For adding location sensing to your Stalker. Connects to the microcontroller via UART. Outputs NMEA Messages.

**RFBee**

**FSK RF Module (CC1101/ATmega168 based)** For adding simple RF capabilities to your Stalker. Onboard microcontroller (Arduino based) runs firmware for RF communication. Connects to the Stalker's microcontroller via UART. Accepts AT Commands.

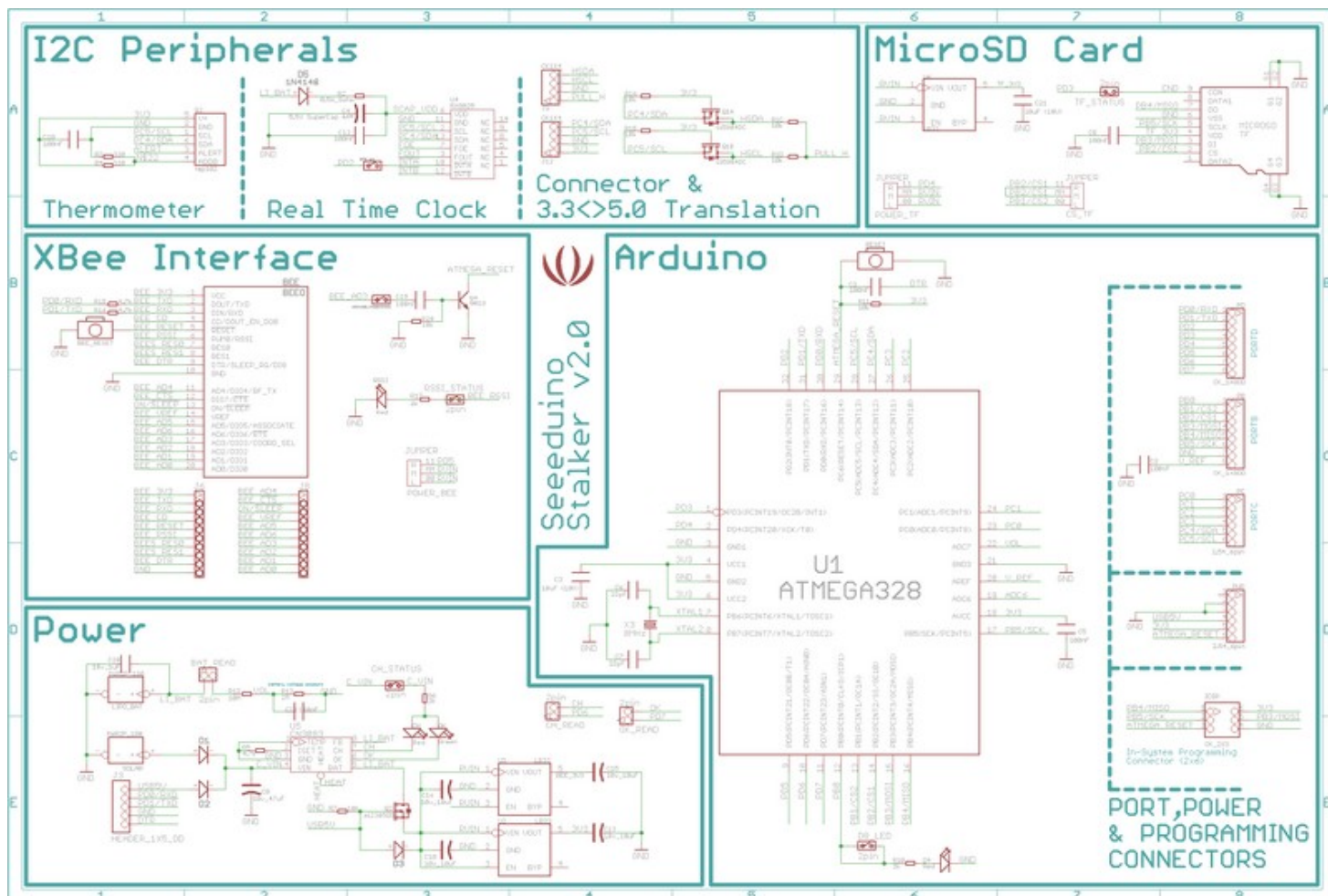


The various Bee modules that can be used with Seeeduno Stalker v2.0 (*Click on the modules to visit their Wiki pages*)

**NOTE:** Please note that [UartSBee](#) cannot be inserted into the Bee Series socket present on Seeeduno Stalker. The [UartSBee](#) is intended for interfacing the other Bee modules (shown in the photo above) to a computer and cannot itself be inserted into the socket meant for these other Bee modules. There is a separate 5 way pinheader present on UartSBee as well as Seeeduno Stalker for interfacing them to each other. This pinheader is composed of: VCC (to supply power to Stalker), TXD, RXD, DTR (for controlling Stalker's Reset signal) & GND.

**Cautions**

- **The product is provided as is without an insulating enclosure. Please observe ESD precautions specially in dry (low humidity) weather.**
- **Please disable bluetooth serial ports when using a Windows based development machine to prevent the Arduino IDE from freezing.**

**Schematics**

[Download v2.0 schematics only \(PDF\)](#)

[Download v2.0b design files \(Cadsoft Eagle\)](#)

NOTE: The Eagle schematics and PCB layout correspond to v2.0b (beta version). There was no change in schematic from v2.0b to v2.0 - only minor improvements were made in the bottom silk screen of the PCB layout.

**Specifications****Key Technical Specifications**

<b>Microprocessor:</b>	ATmega328P
<b>PCB size:</b>	9.27cm x 6.09cm x 0.16cm
<b>Indicators:</b>	Reset, Power, LED on PB0 (Arduino Pin 8)
<b>Power supply:</b>	4.2v Lipo Battery, 5V for charge battery
<b>Power Connector:</b>	2 pin JST/ USB
<b>I/O counts:</b>	20
<b>ADC input:</b>	Dedicated 4 channel (ADC0~ADC3, 10 bit resolution)
<b>Connectivity:</b>	I <sup>2</sup> C, UART, SPI
<b>RoHS:</b>	Yes

**Electrical Characteristics**

Specifications	Min	Norm	Max	Units
Battery JST Input voltage	3.5	4.2	4.2	Volts (DC)
Solar JST Input voltage	4.6	5	6	Volts (DC)
Global Current Consumption	-	300	1000	mA
3.3V I <sup>2</sup> C voltage	3.2	3.3	3.5	Volts (DC)
5.0V I <sup>2</sup> C voltage	4.6	4.7	5	Volts (DC)

UART Baud Rate (while programming)	-	-	115200	bps
---------------------------------------	---	---	--------	-----

## Usage

## Getting Started

If you are new to the "Physical Computing" world and if Seeduino Stalker is the first physical computing platform you want to begin with, then the following steps will help you assemble the hardware and software resources to get you started.

### Step 1: Acquiring the Hardware

Other than a Microsoft Windows based computer, you will require the following hardware for running your first program.



**Seeduino Stalker v2.0**

[Buy Here](#)



**UartsBee v3.1**

Required for programming the Seeduino Stalker.

[Buy Here](#)



**Mini USB Cable**

You would probably have this one lying around, or if not, buy one [here](#). We would use this to connect the UartsBee to one of the USB ports on your computer.



**10 Pin Flat Ribbon Cable (FRC)**

Required for connecting the UartsBee to Seeduino Stalker. Eventhough we require only a 5 way connection, we use this 10 way FRC because its easily available - infact you might already have one lying around your workspace. If not, you can buy a colourful one [here](#).



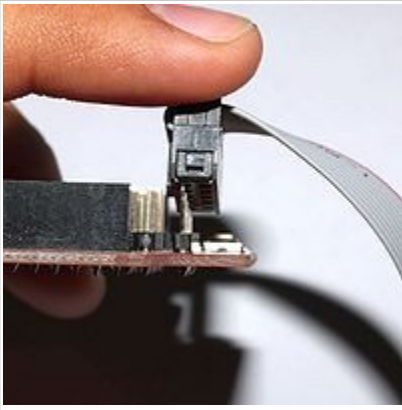
## Step 2: Downloading and installing the software

After you have acquired all of the above hardware, you will need to download and install the software:

1. Visit [www.arduino.cc](http://www.arduino.cc) and go to the download page. There you will find the links to download the latest versions of the arduino software for various operating systems. Download the one for windows - a file named "arduino-xxxx.zip" where xxxx are four digits of version number. So the file would be named something like arduino-0022.zip
2. After you're done downloading it, extract the folder within this zip archive to the root directory of your C: drive.
3. Then just browse over to C:\arduino-0022 using windows explorer and double click on arduino.exe to start it. You can right click on arduino.exe and send a shortcut to it to your desktop for your convenience. (NOTE: If your computer has bluetooth connectivity, make sure to disable any bluetooth serial ports before you start arduino to prevent it from freezing. You can do this by opening the "Device Manager" and uninstalling all the Bluetooth COM ports.)
4. Before you plug-in the hardware to your computer, close arduino.

## Step 3: Installing the drivers and plugging in the hardware

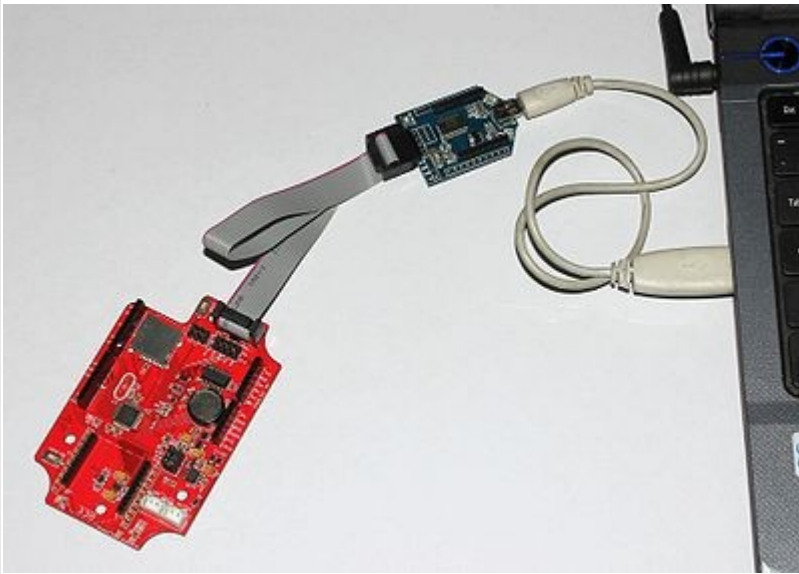
1. UartSBee is like the multi-purpose Swiss Army knife of the Physical Computing world. In our case it will perform three functions:
  - o Allow us to program the Seeeduino Stalker.
  - o Allow us to communicate with Seeeduino Stalker.
  - o Provide power (from USB power of your computer) to Seeeduino Stalker (including any peripherals connected to it).
2. The first two functions of UartSBee (programming and communication) are achieved through an Integrated Circuit called FT232RL which is present on it. Before FT232RL can be used for these purposes, its drivers (certain freely available programs from FT232RL's manufacturer) must be installed on your windows/ubuntu based PC. So before proceeding further, download the driver setup file [from here](#) and install it on your Windows PC.
3. UartSBee has an onboard voltage regulator and a switch to allow you to select what voltage (5.0V or 3.3V) you would like to supply to the target circuit board. In our case the target circuit board is Seeeduino Stalker and so you would need to set this slide switch to 5.0 volts
4. The wiring connection scheme of our hardware is "**Computer** →(Mini USB Cable)→ **UartSBee** →(Flat Ribbon Cable)→ **Seeeduino Stalker**". The Flat Ribbon Cable (FRC) must be connected between UartSBee and Seeeduino Stalker before connecting the UartSBee to the Computer. You can use a single row (only 5 Pins out of the 10 on the cable) of the 10 way FRC to connect the 5 pins on the UartSBee to the 5 pins on the Seeeduino Stalker. Refer the photos below and make sure the signals line up as shown in the table (**Note: The TXD and RXD pins must be cross connected as shown in the table**).
5. Next connect the Mini USB cable from UartSBee to your computer. If you are using a Windows based PC, the "Found New Hardware" balloon will popup and within a few moments the drivers for FT232RL (i.e. UartSBee) will be installed.



Connect the Flat Ribbon Cable to Stalker's UART Programming header.



Connect the other end of the Flat Ribbon Cable to the header on UartSBee



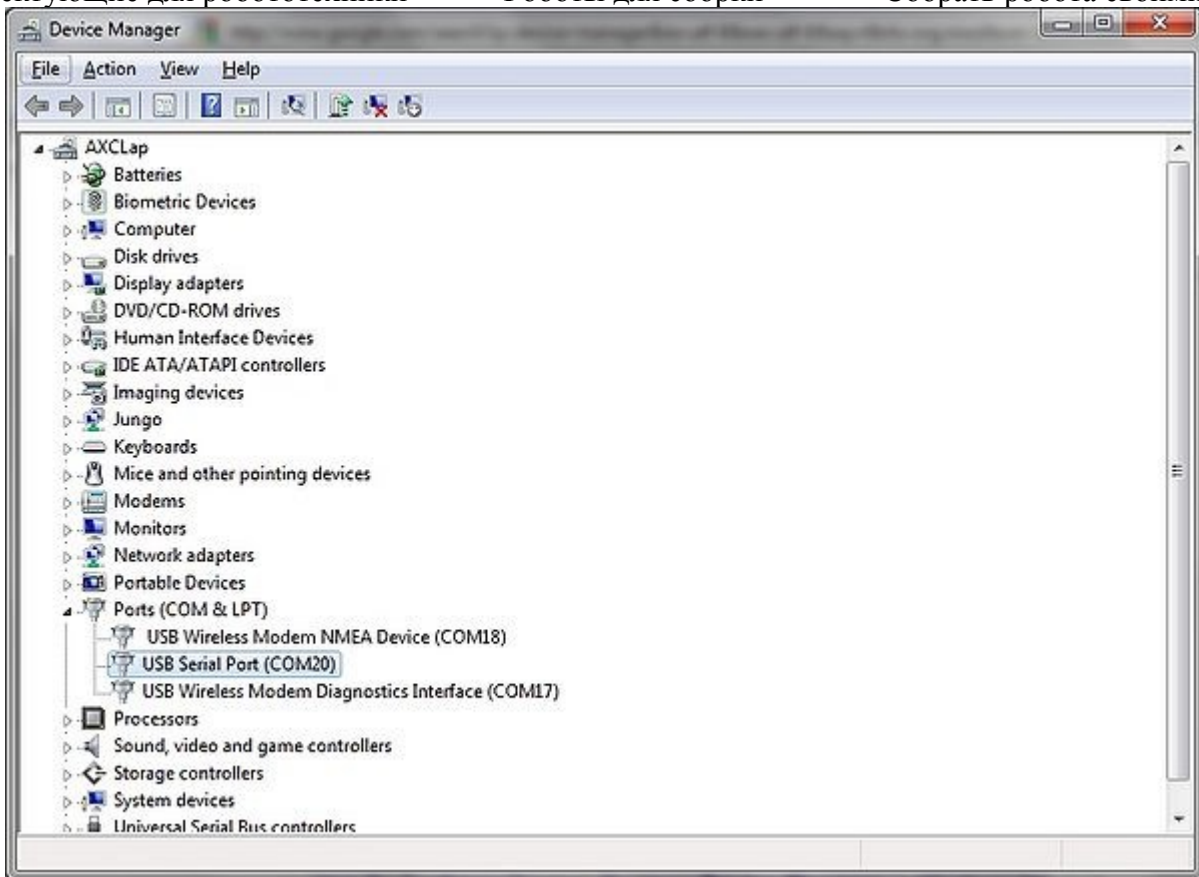
Complete the setup by plugging in the UartSBee to a USB port on your computer using the Mini USB cable. Make sure the voltage select switch on UartSBee is set to 5.0V and not 3.3V.

Flat Ribbon Cable Connection table

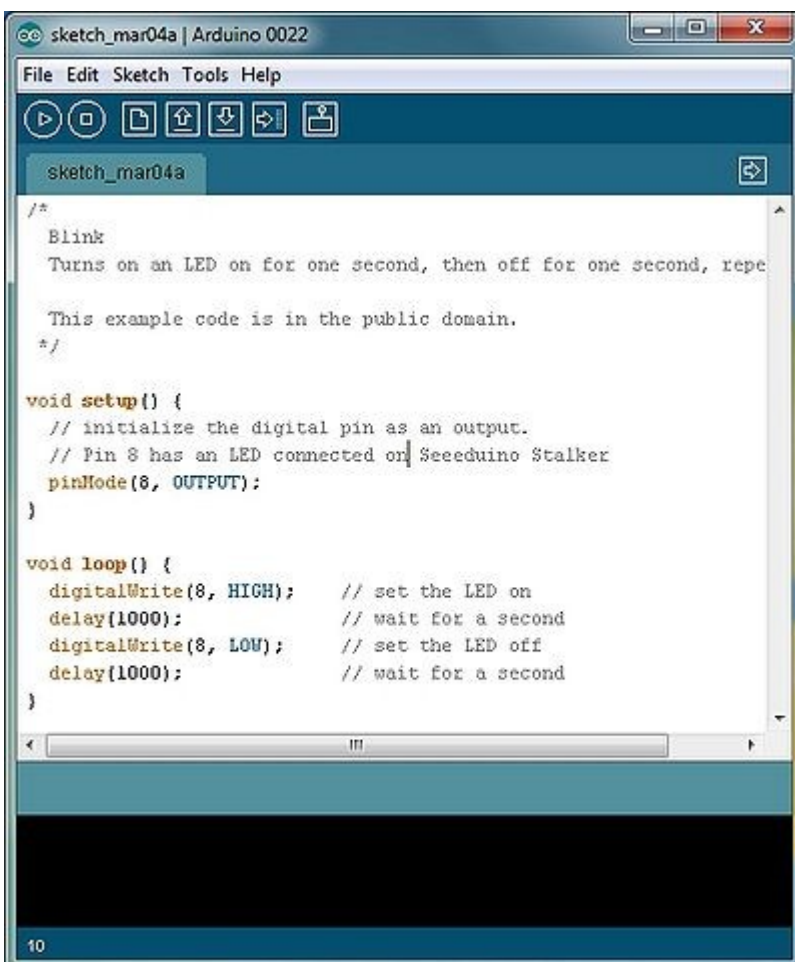
Seeduo Stalker		UartSBee
USB5V	↔	VCC
RXD	↔	TXD
TXD	↔	RXD
GND	↔	GND
DTR	↔	DTR

#### Step 4: Creating a new sketch for blinking the LED and downloading it to the board

1. Before proceeding further, we need to find out what COM port number has been assigned to the UartSBee. To do this go to the "Device Manager" (Press Windows key and the R key on your keyboard simultaneously, then when the Run dialog shows up, type "devmgmt.msc" into it and press enter) and look under the category "Ports (COM & LPT)". You will find an entry "USB Serial Port" as shown in the screenshot below. Note the COM port number next to it - for example in the screenshot below its its COM20.



- Next, start Arduino by double-clicking the shortcut you created for it on your desktop in step 1 above. A blank sketch with the today's date as its name will be created for you. Into this sketch type the code for blinking the User LED on Seeeduino Stalker v2.0 as shown below:



```

/*
  Blink
  Turns on an LED on for one second, then
  off for one second, repeatedly.

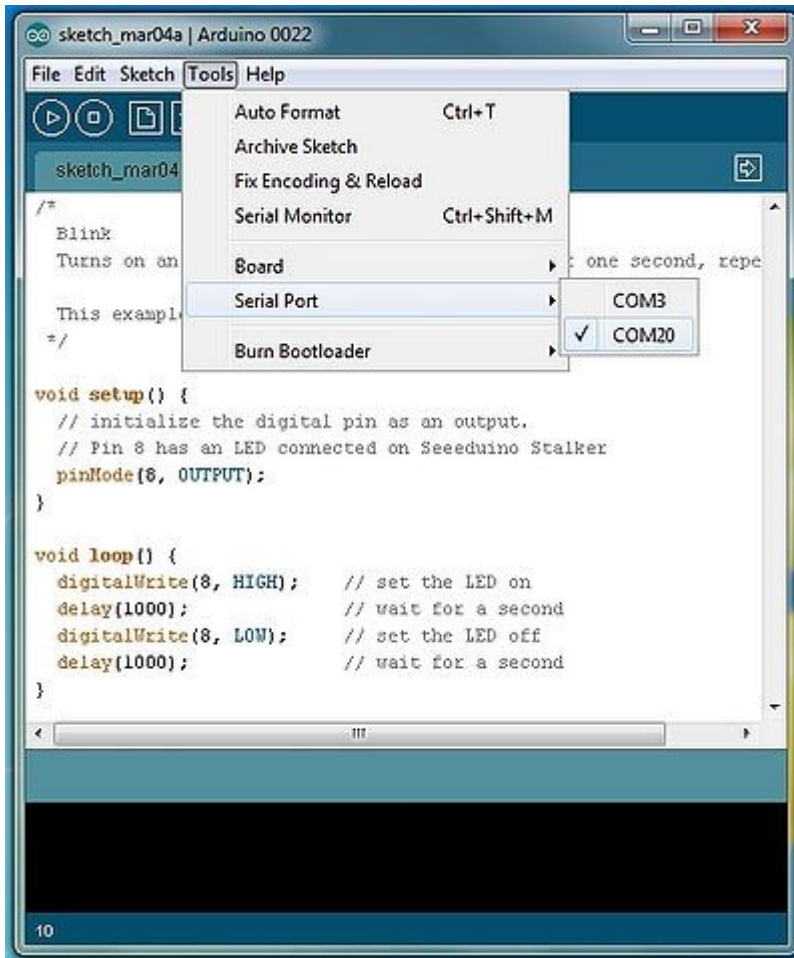
  This example code is in the public
  domain.
  */

void setup() {
  // initialize the digital pin as an
  output.
  // Pin 8 has an LED connected on
  Seeeduino Stalker
  pinMode(8, OUTPUT);
}

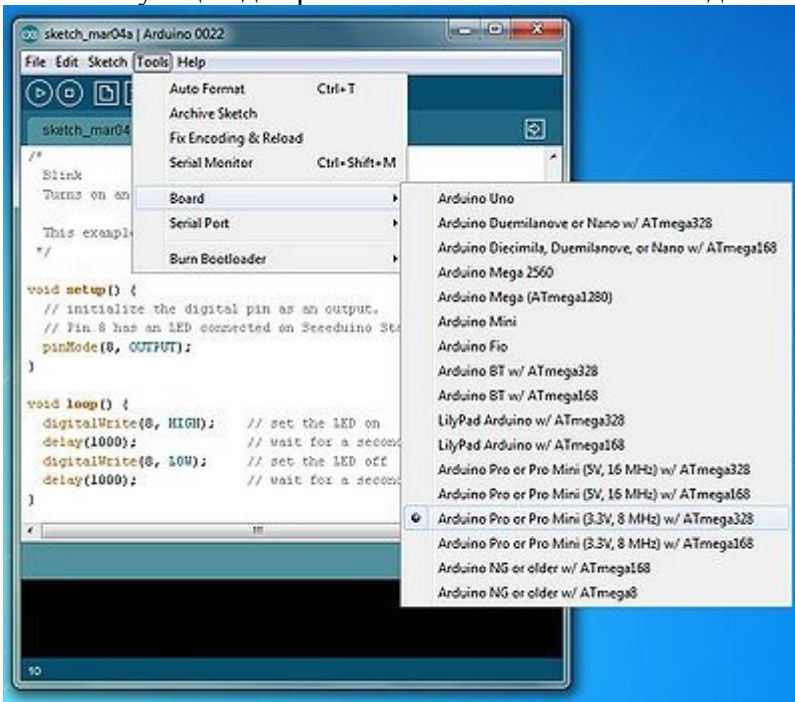
void loop() {
  digitalWrite(8, HIGH); // set the
  LED on
  delay(1000);           // wait for a
  second
  digitalWrite(8, LOW); // set the
  LED off
  delay(1000);          // wait for a
  second
}
  
```



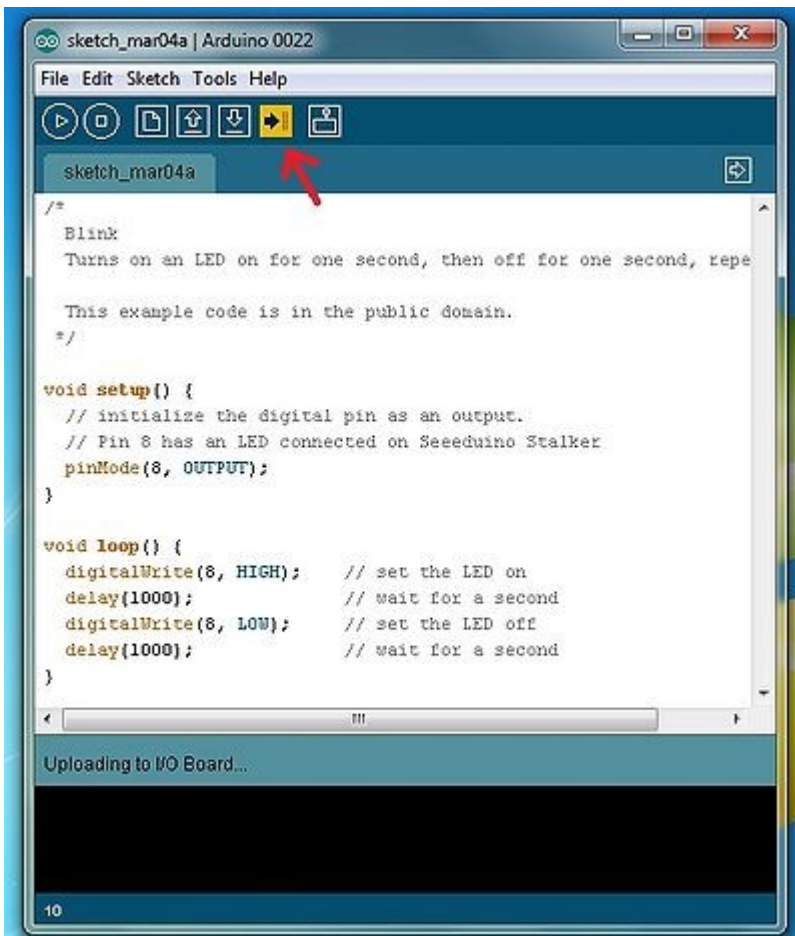
1. Before we can program the Seeeduino Stalker, we need to tell Arduino the Serial Port number of our UartSBee/Stalker. We can do this by going to **Tools>Serial Port** and selecting COM port that we had obtained in the previous step.



2. Arduino also needs to know what kind of microcontroller and circuit board we are using. Since Seeeduino Stalker v2.0 is based on the Duemilanove board and has an ATmega328P, select the corresponding option from the **Tools>Board** menu and select "**Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328**". (Please note that even though UartSBee is supplying 5.0V to the Seeeduino Stalker board, the microcontroller (ATmega328P) itself is operating at 3.3V. Voltage regulator on board the Seeeduino Stalker converts 5.0 volts power supply to 3.3 volts.)



1. Now we are ready to upload the program into our Seeeduino Stalker. To do this click on the "Upload" button as shown below and within a few moments, the LED on Seeeduino Stalker will start blinking. The RX and TX LEDs on UartSBee will flicker while the programming is being carried out.



Since all the required software is now already installed on your computer, the next time you want to create and upload a new sketch into Seeeduino Stalker, you just need to connect it to the Computer via UartSBee and repeat Step 4.

## Connection Notes

- **Bee series socket** - 2\*10 pin 2.0mm pitch (which will mate with - one at a time - any of the [wireless modules](#): [XBee](#), [BluetoothBee](#), [GPSBee](#) or [RFBee](#).) Communication with these modules are done through UART.
- **Serial interface** – To save space and lower costs, USB<->Serial connectivity is not provided by default. You may use the FT232 based [UartSBee](#) or other USB to serial adapter boards to do the programming or communicate with the PC.
- **User LED** – An LED has been provided onboard for use in your application as desired.
- **I<sup>2</sup>C Interface**: Onboard I<sup>2</sup>C level shifter IC provides voltage translation between 3.3V and 5V devices. This allows you to connect 5.0 Volt I<sup>2</sup>Cs to you microcontroller when its operating on 3.3 Volts.

## Jumpers

### Battery Related

- **BAT\_READ** (*Jumper type: 2 pin, Location: **Top**, Factory state: **mounted***)  
If a jumper is placed on BAT\_READ, the Lithium Battery voltage can be read via the analog pin 7 (ADC7) of ATmega328P. This pin is only available on the ATmega328P SMD package which is present on Seeeduino Stalker. (The Dual-Inline Package (DIP) only has analog pins from 0 thru 5.)
- **CH\_READ** (*Jumper type: 2 pin, Location: **Top**, Factory state: **mounted***)  
If a jumper is placed on CH\_READ, the Lithium Battery charger IC's (CN3083 or CN3063) CHRГ (or CH) signal will be connected to digital pin 6 (PD6) of ATmega328P. It is an open drain signal which will be pulled low "while" the battery is charging. It will be left in high impedance state if there is a charging error. A Red SMD LED is also connected to this signal. This signal is not affected by the make or break of the CH\_STATUS jumper.
- **OK\_READ** (*Jumper type: 2 pin, Location: **Top**, Factory state: **mounted***)  
If a jumper is placed on OK\_READ, the Lithium Battery charger IC's (CN3083 or CN3063) DONE (or OK) signal will be connected to digital pin 7 (PD7) of ATmega328P. It is an open drain signal which will be pulled low "after" the battery has been fully charged. It will be left in high impedance state while the charging cycle is incomplete or when charging is not taking place. A Green SMD LED is also connected to this signal. This signal is not affected by the make or break of the CH\_STATUS jumper.
- **CH\_STATUS** (*Jumper type: **Solder bridge**, Location: **Top**, Factory state: **connected by a thin track***)  
Two LEDs present on board Seeeduino Stalker indicate the charging status of the LiPo Battery. The Red SMD LED is connected to the CHRГ (or CH) pin of the Lithium Battery charger IC (CN3083 or CN3063) which when glowing, indicates that the charging cycle of the battery is active. The Green SMD LED is connected to the DONE (or OK) pin of the Lithium Battery charger IC (CN3083 or CN3063) which when glowing, indicates that the battery is now fully charged. When operating Seeeduino Stalker out in the field, you can cut the track connecting the CH\_STATUS pads to disconnect these LEDs from C\_VIN and hence conserve power which would have otherwise been consumed in lighting these LEDs.

### Real Time Clock (RTC) Related

- **INTA\_RTC** (*Jumper type: **Solder bridge**, Location: **Bottom**, Factory state: **NOT connected by a thin track***)  
Close this jumper to allow the Real Time Clock (RTC) to wake the ATmega328P at a specified time. This jumper will connect INTA of the RTC Chip (RX8025) to PD2 (INT0) of ATmega328P.

### microSD Card (TransFlash Card) Related

- **CS\_TF** (*Jumper type: **Solder bridge - 2 way**, Location: **Bottom**, Factory state: **SS and PB2 connected by a thin track***)  
This jumper is a two way jumper made up of three pads: PB1, SS and PB2. SS is the Chip Select signal of



the the microSD card. By default SS is connected by a thin track to PB2 - Digital Pin 10 (PB2) of the microcontroller. If instead you want to connect the Chip Select signal from the microSD card to Digital pin 9 (PB1), just cut the track between PB2 and SS and put a solder blob between SS and PB1.

- **TF\_STATUS** (*Jumper type: Solder bridge, Location: Bottom, Factory state: NOT connected by a thin track*)

If you close this solder jumper, Digital Pin 3 (PD3) of the microcontroller will go low when there is a card inserted in the microSD card socket. This can be used for card detection.

- **POWER\_TF** (*Jumper type: Solder bridge - 2 way, Location: Bottom, Factory state: EN and VCC connected by a thin track*)

This jumper is a two way jumper made up of three pads: VCC, EN and CTL\_D4. EN is the TF power regulator Enable pin. By default EN is connected by a thin track to VCC to always enable TF power. If instead you want to Control the TF Card power With Digital Pin 4 (PD4) of the microcontroller, just cut the track between EN and VCC and put a solder blob between EN and CTL\_D4. you can control TF Card Power by Digital Pin 4 (PD4), you would be able to completely turn off the microSD card to conserve power when operating out in the field.

## Bee Module Related

- **POWER\_BEE** (*Jumper type: Solder bridge - 2 way, Location: Bottom, Factory state: EN and VCC connected by a thin track*)

This jumper is a two way jumper made up of three pads: VCC, EN and CTL\_D5. EN is the XBee power Enable pin. By default EN is connected by a thin track to VCC to always enable XBee Power. If instead you want to Control XBee Power by Digital Pin 5 (PD5) of the microcontroller, just cut the track between EN and VCC and put a solder blob between EN and CTL\_D5. You can control XBee Power by Digital Pin 5 (PD5), you would be able to completely turn off the Bee module to conserve power when operating out in the field.

## Bee Module Related - Only XBee related

- **WIRELESS\_PROGRAMMING** (*Jumper type: Solder bridge, Location: Bottom, Factory state: Connected by a thin track*)

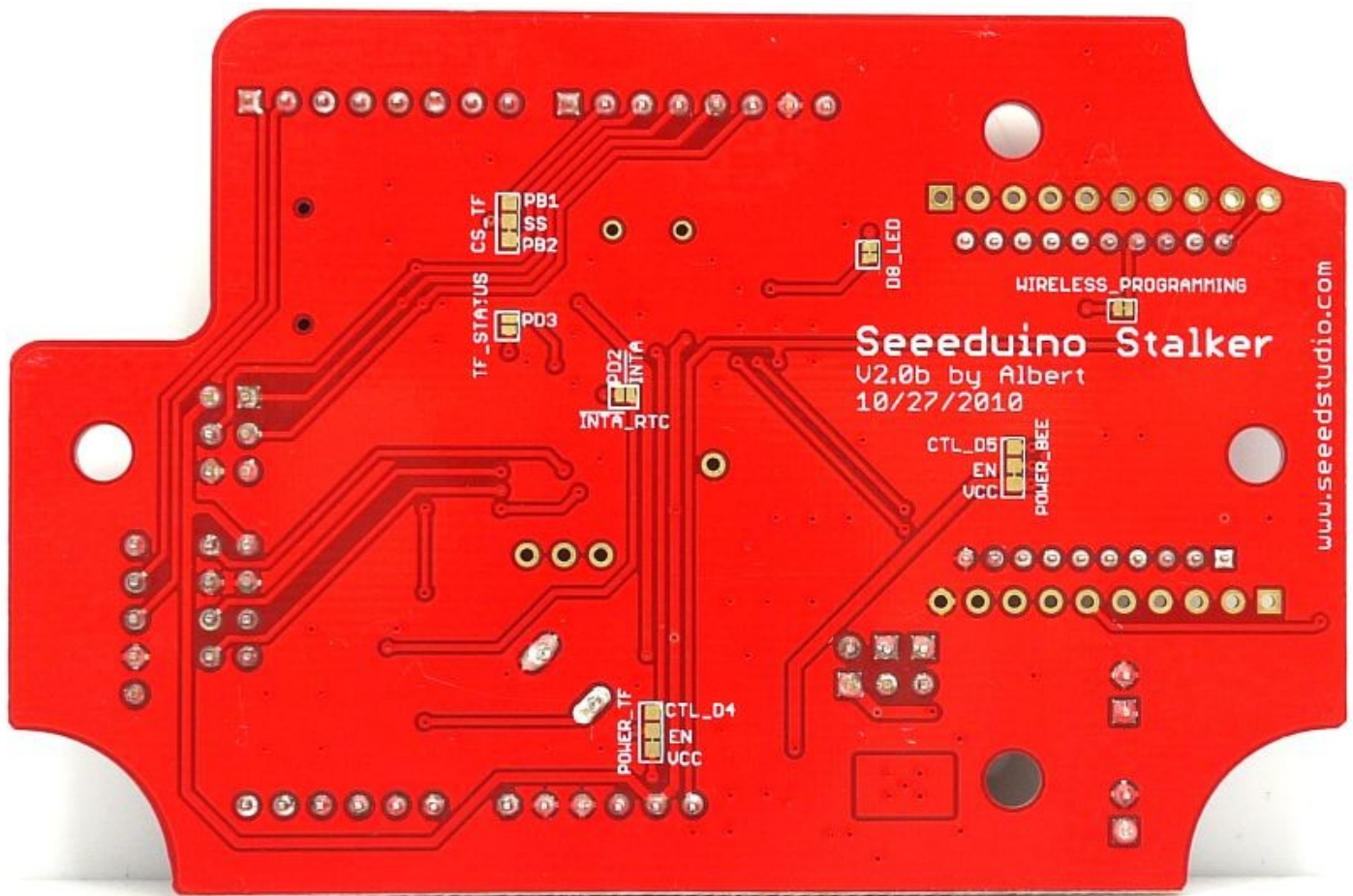
You can use Digi's XBee modules to wirelessly program the ATmega328P on your Seeeduino Stalker. An XBee module must be configured and installed on your Seeeduino Stalker and another XBee module must be connected to your Laptop via a UartSBee. The pin DIO3 on the Seeeduino Stalker will be used to control the Reset Pin of ATmega328P. This jumper allows you to enable or disable (default: enabled) the control of the Reset Pin of ATmega328P by the DIO3 pin of the XBee module. You can cut the track between the pads of this jumper if you don't want the DIO3 pin to control the Reset pin of ATmega328P. Lady Ada has a nice [tutorial](#) on how to remotely program your Arduino based product using XBee. (NOTE: Both the XBee - the one on Stalker and the one connected to the PC must be pre-configured once using the X-CTU software before use.)

- **RSSI\_STATUS** (*Jumper type: Solder bridge, Location: Top, Factory state: connected by a thin track*)

A red LED present on the top side of the PCB is connected to the RSSI (Received Signal Strength Indicator) pin of the XBee module. XBee outputs a PWM signal on this pin which is directly proportional to the quality of the RF link when the last packet was received by it. This PWM signal when applied to the LED would vary its brightness as per the quality of the RF link - better the link, brighter the LED. Since this LED would consume power, you can cut the track between the pads of this jumper to conserve battery power out in the field. RSSI value is also available over the UART using the DB command (measured in -dBm). (NOTE: The DB value only indicates the received signal strength of the last hop. If a transmission spans multiple hops, the DB value provides no indication of the overall transmission path, or the quality of the worst link – it only indicates the quality of the last link and should be used sparingly.)

**User LED Related**

- **D8\_LED** (*Jumper type: Solder bridge, Location: Bottom, Factory state: connected by a thin track*)  
A red LED present on the top side of the PCB is connected to the Digital Pin 8 (PB0) of the microcontroller. The LED is free for use by you in your program for any purpose you desire. Since this LED would consume power, you can cut the track between the pads of this jumper to conserve battery power out in the field and use Digital Pin 8 for other purposes like controlling a shield.



The bottom side of Seeeduino Stalker v2.0 showing. Note the location of various jumpers.

**External Crystal**

- The Seeeduino Stalker is supplied with ATmega328P's internal fuse bits programmed for operation at 8 MHz using the internal RC oscillator (a bit inaccurate but conserves power). Alternatively, if you would like to use an external crystal (more accurate), then there is a footprint provided for mounting it on the PCB between the XTAL1 and XTAL2 pins of the microcontroller. Two 22pF capacitor (corresponding to the value required for an 8MHz crystal) have already been mounted. After mounting the crystal, you will need to reconfigure the fuse bits using an In-System Programmer to get the microcontroller to operate using this external crystal.

**Datasheets of Components**

- [ATmega328P](#) - 8-bit AVR Microcontroller present onboard Seeeduino Stalker.
- [CN3063](#) - Charger controller for Lithium batteries (charging using solar panel)
- [Si5904DC](#) - Dual N-Channel 2.5 V (G-S) MOSFETs (used for 3.3V <math>\leftrightarrow</math> 5.0V translation for I<sup>2</sup>C, refer NXP's [AN97055](#)).
- [Si2305DS](#) - P-Channel 1.25-W, 1.8-V (G-S) MOSFET.
- [TMP102](#) - Low power digital temperature sensor with I<sup>2</sup>C interface.
- [RX-8025](#) - Precision Real Time Clock (RTC) Module with I<sup>2</sup>C interface. (An application manual from Epson Toyocom is [here](#)).

## Source Code Examples

A sketch demonstrating the use of various peripherals present onboard Seeeduino Stalker is attached below. It logs the date/time from the RX8025, temperature from TMP102 and battery voltage into a text file which is stored on the microSD card. It also sends this data to the serial port. You can copy various parts of this code to kickstart the coding for your own project.

Please note that the demonstration sketch presented below makes use of the fat16lib Arduino Library. Please visit [fat16lib's project page](#) to obtain the latest version of the library. The library source files will be within a zip file named something like "fat16lib20101009.zip". Within this zip file you will find - amongst other things - a folder named Fat16. Extract this folder and place it as it is into your Arduino installations library folder - i.e. if you followed the [Getting Started](#) guide above, you will need to place the Fat16 folder in "C:\arduino-0022\libraries". A copy of this Fat16 folder can also be found on our own server here: [File:Stalker code Fat16.zip](#). Please do install the fat16lib before opening the demonstration sketch.

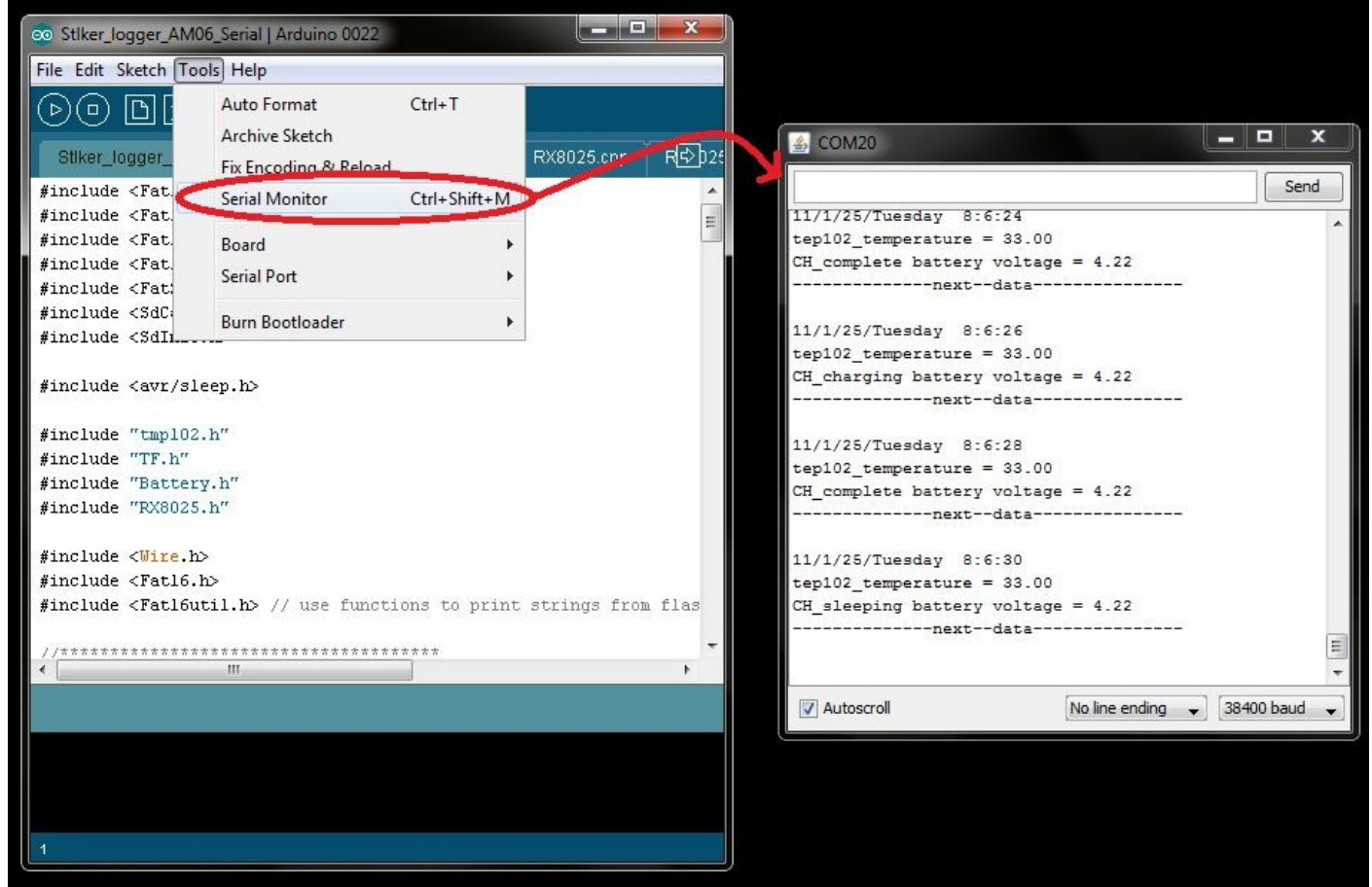
Few notes about the demonstration sketch:

- The data is sent to the serial port at 38400 bps 8-N-1. The data is also logged to the microSD card in a named "DATA.LOG".
- DATA.LOG is a text file and can be opened using the notepad application on Windows. To do this you need to remove the microSD card from Seeeduino Stalker and insert it into a [card reader](#) and connect it to the USB port of the computer. The card will show up as a removable drive withing Windows Explorer and you will be able to access the DATA.LOG file.
- The microSD must be a standard SD card and not a SDHC card. fat16lib does not support SDHC cards.
- The microSD card must be pre-formatted with FAT16 file system. You can use a windows PC to do this.
- To view the data sent live to the UART, you will need a [UartSBee](#) setup as explained in the [Getting Started](#) guide above. You can use the "Serial Monitor" which is a part of the Arduino IDE to view this data.

Seeeduino Stalker demonstration sketch: [File:Stlker logger AM06 Serial.zip](#)

(This archive file contains a folder named "Stlker logger AM06 Serial". Extract and place this folder in your Arduino sketch folder which is usually the folder named "Arduino" present in your My Documents. Start Arduino IDE and open the "Stlker logger AM06 Serial.pde" sketch.)





Using the Arduino IDE's "Serial Monitor" to view the data sent out its UART by a Seeeduino Stalker running the demonstration sketch

## Accessories

Besides the [various Bee Series modules](#), the following accessories are available for use with Seeeduino Stalker v2.0:



Water-proof Enclosure  
for Seeduino Stalker v2.0



SanDisk 2GB  
microSD Card



microSD Card Reader  
(in a capsule)

Note: If you buy the "microSD Card Reader (in a capsule)", you would not need the "Mini USB cable" since the former also doubles up as a Mini USB cable. Check out microSD Card Reader (in a capsule)'s [Wiki page](#) for



Mini USB cable for use with  
UartSBee while programming



10 pin Flat Ribbon Cable for use with  
UartSBee while programming



Lithium Polymer (LiPo)  
Battery  
clarification.



Solar Panel for  
charging LiPo Battery via Stalker

## FAQ

Here is the Seeduino Stalker FAQ, users can list the Frequently Asked Questions here, example as below:

1. There is no coin-cell battery present on the version 2.0 of Seeduino Stalker, instead there is a super capacitor. Have you evaluated/calculated the amount of time the RTC can run after power has been removed?

During our tests, we found out that it will keep the RTC ticking for at least 3 days. Super capacitors are better than coin cells in the sense that there is no memory effect and there is no hassle of replacing or recharging the coin cell.

2. Has the user defined switch been removed on version 2.0 of Seeduino Stalker?

Yes, we have removed the user button on the version 2.0

3. In version 1.0 of Seeduino Stalker the microcontroller could be woken up from sleep mode via interrupt from the Bee module. Is this same feature also available on version 2.0?

In the previous version (i.e. 1.0), the microcontroller could be woken on RF data packet reception by the Bee module via its pin 15 which was connected to INT0 (PD2) of the microcontroller. In the newer version, the INT0 (PD2) pin of the microcontroller is connected by a jumper ([INTA\\_RTC](#)) to the INTA pin of the RTC chip which can wake it up at a pre-configured time (or periodically). Since Seeeduino Stalker is meant for use as a wireless sensor network node, this modification would be useful in cases where the microcontroller must wake up periodically to transmit sensor readings and go to sleep again.

4. The I<sup>2</sup>C pin headers on Version 1.0 of Seeeduino Stalker allowed easy connection to external 3.3V as well as 5.0V devices. Is the same feature available on the version 2.0?

Yes, in fact we have improved it - previously PCA9306 was used for level translation on I2C bus. But now we use N channel MOSFETs for translation - this technique has many advantages (refer NXP's Application Note [AN97055](#)).

5. There is no English datasheet for the CN3083, what do I do?

We had used CN3083 on the beta revision of version 2.0 of Seeeduino Stalker (v2.0b). The final v2.0 version will have CN3063 on it. The datasheet for CN3083 is only available in the Chinese language. On the other hand, CN3063 has an English language datasheet and is attached above. Both parts: CN3063 as well as CN3083 are very much similar in operation.

6. I find the explanation of the battery related jumpers given above a bit confusing, I need a simpler explanation.

**BAT\_READ** - Allows you to read the battery voltage via Analog Pin 7 of the microcontroller using its builtin Analog to Digital Convertor.

**CH\_READ** - Digital Pin 6 of the microcontroller will remain low (logic 0) while the battery *is being* charged. Unmount this jumper to use Digital Pin 6 for other purposes.

**OK\_READ** - Digital Pin 7 of the microcontroller go low (logic 0) after the battery *has been* charged. Unmount this jumper to use Digital Pin 7 for other purposes.

**CH\_STATUS** - The above two signals (CH\_READ and OK\_READ) also have LEDs connected to them. CH\_STATUS jumper allows you to disable these LEDs and decrease power consumption.

## Support

If you have questions or other better design ideas, you can go to our [forum](#) or [wish](#) to discuss.

## Revision History

Revision	Descriptions	Release Date
<a href="#">Seeeduino Stalker v1.0</a>	Initial public release	Dec 23, 2009
<a href="#">Seeeduino Stalker v2.0b</a>	Beta of New and improved version with more features	Dec 17, 2010
<b>Seeeduino Stalker v2.0</b>	Final non-beta (minor improvements on bottom silkscreen)	Mar 14, 2011

## Issue Tracker

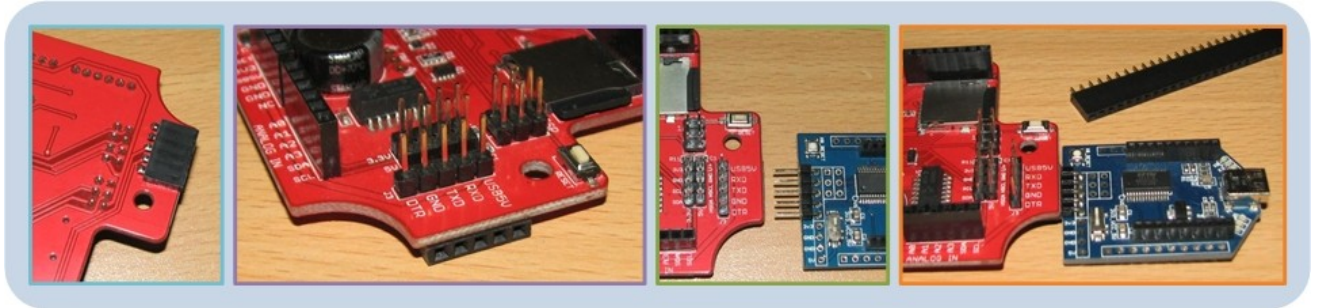
Issue Tracker is the place you can publish any suggestions for improvement or any bugs you think you might have found during use. Please write down what you have to say, your suggestions will help us improve our products.

1. **UartSBee Female header** (Issue Type: Improvement)

The next version of Seeeduino Stalker may incorporate the following improvement. Alternatively, the users who own Seeeduino Stalker v2.0 can perform this modification themselves. It greatly simplifies the process of connecting a UartSBee to Seeeduino Stalker v2.0. *This modification is especially useful when*



*using Seeeduino Stalker with an Arduino compatible shield - there won't be any space between the two stacked boards to install a connector over the 5 way pin header meant for programming and communication via UartSBee.*



Modifying Seeeduino Stalker v2.0 by adding a 5 pin female header to allow UartSBee to connect to it easily without the use of any cable.

## **Additional Ideas**

The Additional Ideas is the place to write your project ideas about this product, or other usages you've found. Or you can write them on Projects page.

## **Resources**

Cadsoft Eagle format source file: [http://www.seeedstudio.com/wiki/images/7/7f/Seeeduino\\_Stalker\\_v2.0b.zip](http://www.seeedstudio.com/wiki/images/7/7f/Seeeduino_Stalker_v2.0b.zip)

R8025 RTC, tmp102 libraries and power save demo codes, **AND you need solder a bridge on INA\_PD2 solder jumper:**

[http://www.seeedstudio.com/wiki/images/c/cb/Seeeduino\\_Stalker\\_V2.0\\_R8025\\_and\\_TMP102\\_library.zip](http://www.seeedstudio.com/wiki/images/c/cb/Seeeduino_Stalker_V2.0_R8025_and_TMP102_library.zip).

## **How to buy**

Seeeduino Stalker v2.0 can be ordered through the Bazaar. Its product page is located [here](#).

## **See Also**

- [Seeeduino Stalker v1.0](#)
- [Comparison between various versions of Seeeduino Stalker](#)
- [Seeeduino Stalker v2.0 Enclosure](#)

## **Licensing**

This documentation is licensed under the Creative Commons [Attribution-ShareAlike License 3.0](#) Source code and libraries are licensed under [GPL/LGPL](#), see source code files for details.