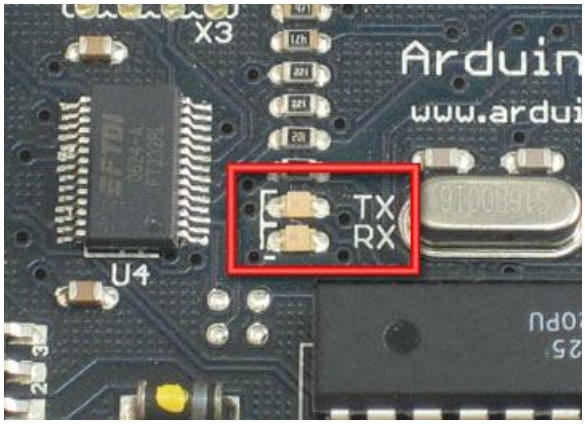
Serial Communication

SoftwareSerial Library

The Arduino SoftwareSerial library enables you to create a separate (virtual) Serial communication line on any Arduino I/O pin. With a max speed of 9600bps, SoftwareSerial is suitable for sending commands from the Arduino to the Sabertooth motor controller. The Sabertooth does not send any information back to the Arduino, so you only need a Serial transmit pin (Tx); although you set up both.

Introduction

The Serial library can be used to communicate between Arduino(s) and computers over a USB cable (wired) or** Bluetooth/Zigbee (wireless). **This inbuilt library contains built-in functions for the same. In fact, it is used implicitly when we upload sketches/programs to the Arduino. There are two data lines in serial communication, RX and TX, stand for receiver and transmitter. A communication wire, e.g. USB typically connects the RX pin of the Arduino to the TX pin of the computer; and the TX pin of the Arduino to the RX pin of the computer. The board has a couple of LEDs which light up when data is being received / transmitted.



Basic Communication

If you face issues while uploading the code, make sure that pins 0 and 1 of the Arduino are not connected to anything, since they double up as RX and TX pins.

The **Serial.begin() **function starts off the communication channel. The 9600 parameter is the data rate in bits per second, which must be the same for both the receiver and the sender.

The Serial.println() function takes in a string or variable and transmits it, followed by a new line. Upload the sketch on the board, and press Ctrl+Shift+M to bring up the Serial monitor.

The monitor displays all incoming data from the board, as well as allows sending data to it. The** Serial.read() **function returns the last received character by the Arduino, and is illustrated in the next example.

Another useful function is Serial.available() which returns true at the moment the Arduino receives data. It can be used effectively inside a loop to check for incoming commands.

//code

```
#include <SoftwareSerial.h> // Tell Arduino to use the SoftwareSerial library
// define the transmit and receive pins to use
#define rxPin 2
#define txPin 3
// set up a new serial port to use pins 2 and 3 above
SoftwareSerial mySerial = SoftwareSerial(rxPin, txPin);
void setup() {
Serial.begin(9600); // start the standard Serial monitor
// define pin modes for SoftwareSerial tx, rx pins:
pinMode(rxPin, INPUT);
pinMode(txPin, OUTPUT);
// set the data rate for the new SoftwareSerial port
mySerial.begin(9600);
}
void loop() {
mySerial.print(0, BYTE); // this will print through pin 3 to the Sabertooth
Serial.print("Hello"); // this will print through pin 1 to your computer Serial monitor
delay(100);
}
```

For more information regarding the uses and limitations of the SoftwareSerial library, please visit the Arduino reference page at www.arduino.cc/en/Reference/SoftwareSerial. Now that you have the Arduino talking to the Sabertooth, consider what it needs to say to the Sabertooth to make the motors move.