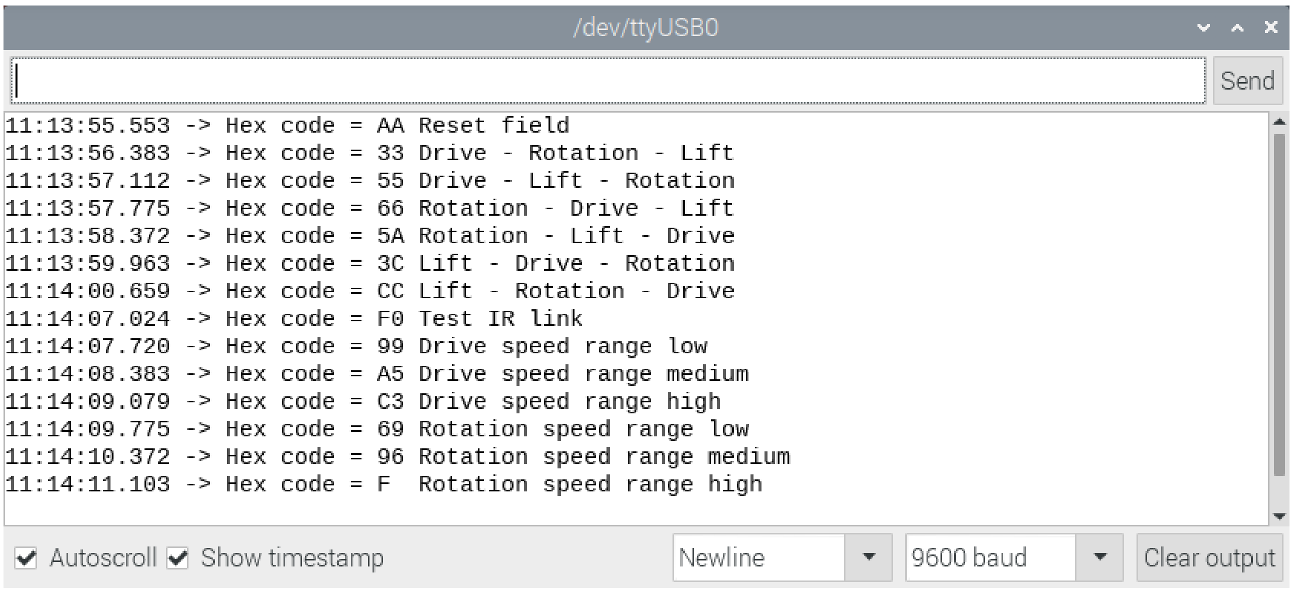
IR Code Viewer

For the 2022 BEST game “Made 2 Order”

This is a simple program that works with an Arduino Uno plus a BEST IR RCVR to display the single byte codes transmitted by a team robot using the BEST IR XMTR. Use it to see the codes your team’s robot is actually sending and verify that the IR Transmitter to Receiver link is working.

The program will receive the codes and display them on the Arduino IDE’s Serial Monitor like this:



```
11:13:55.553 -> Hex code = AA Reset field
11:13:56.383 -> Hex code = 33 Drive - Rotation - Lift
11:13:57.112 -> Hex code = 55 Drive - Lift - Rotation
11:13:57.775 -> Hex code = 66 Rotation - Drive - Lift
11:13:58.372 -> Hex code = 5A Rotation - Lift - Drive
11:13:59.963 -> Hex code = 3C Lift - Drive - Rotation
11:14:00.659 -> Hex code = CC Lift - Rotation - Drive
11:14:07.024 -> Hex code = F0 Test IR link
11:14:07.720 -> Hex code = 99 Drive speed range low
11:14:08.383 -> Hex code = A5 Drive speed range medium
11:14:09.079 -> Hex code = C3 Drive speed range high
11:14:09.775 -> Hex code = 69 Rotation speed range low
11:14:10.372 -> Hex code = 96 Rotation speed range medium
11:14:11.103 -> Hex code = F  Rotation speed range high
```

✓ Autoscroll ✓ Show timestamp — Newline — 9600 baud

Hardware Required

To use this program you will need the following hardware:

- A PC running the Arduino IDE
- An Arduino Uno
- A BEST IR RCVR (you built this alongside your XMTR but don’t need it on your robot)
- 3 male – female breadboard wires (to connect BEST IR RCVR to Arduino Uno)
- A USB cable to connect the Arduino Uno to your PC

Circuit Hookup

Connect the PC, Arduino Uno, and BEST IR RCVR together as follows:

- Use a breadboard wire (ideally black) to connect any GND of the Uno to the IR RCVR pin closest to the large blue capacitor.
- Use another breadboard wire (ideally red) to connect the next pin on the IR RCVR to any 5V pin of the Uno.
- Use the third breadboard wire (ideally a third color) to connect the third pin on the IR RCVR to Digital connection 10 of the Uno.
- The fourth pin on the IR RCVR, farthest from the large blue capacitor, remains unconnected.
- Use the USB cable to connect the Uno to your PC.

Load and Use the Program

Cut & paste the program from the Appendix into a new project in the Arduino IDE. From there upload it to the Uno. It should start running in a few seconds.

To see the output you will need to open the serial monitor. In the Arduino IDE this is menu item "Tools" –> "Serial Monitor". Make sure the serial monitor is set to 9600 baud.

Now aim the IR XMTR on your robot at the IR RCVR sensor. As your robot sends codes they should be displayed in your PC's serial monitor window.

Appendix: The Program

```
/*
 23Feb22  Written by Steve Marum
         Target device is Uno

 For team use testing their robot's IR transmitter.
 Receives IR signals using the BEST IR RCVR.
 Flashes LED_BUILTIN when a character is received over the IR link.
 Displays all codes received by the IR RCVR on the Arduino's serial monitor.
 Shows hex code and description of the code.
 Invalid codes are also displayed and flagged.
 */

//  We are using SoftwareSerial to receive the IR signal.
//  The built-in serial port is used to send to the Arduino IDE serial monitor.
#include <SoftwareSerial.h>
#define rxPin 10
#define txPin 11
SoftwareSerial IRreceive(rxPin, txPin);  //  Note: only certain Mega pins can be used.

// variables will change:
int IRrcv;                  // variable to hold character received via IR link

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);

  // initialize software UART pins & initialize SoftwareSerial
  pinMode(rxPin, INPUT_PULLUP);
  pinMode(txPin, OUTPUT);
  IRreceive.begin(600);

  // initialize console serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  // Turn on LED_BUILTIN if a byte is received
  IRrcv = IRreceive.read();
  if (IRrcv != -1) digitalWrite(LED_BUILTIN, HIGH);   // LED on
  else digitalWrite(LED_BUILTIN, LOW);              // LED off
  // Flush the serial buffer.  Read buffer until -1 (empty signal) is detected
  while (IRreceive.read() != -1);

  delay(10);   // This seems to be needed for SoftwareSerial to work correctly!?!
  if (IRrcv != -1) {
    Serial.print("Hex code = ");
    Serial.print(IRrcv, HEX);
    Serial.print("   ");
    switch (IRrcv) {
      case 0xAA:
        Serial.println("Reset field");
        break;
      case 0xF0:
        Serial.println("Test IR link");
        break;
      case 0x33:
        Serial.println("Drive - Rotation - Lift");
        break;
      case 0x55:
        Serial.println("Drive - Lift - Rotation");
        break;
      case 0x66:
        Serial.println("Rotation - Drive - Lift");
        break;
      case 0x5A:
        Serial.println("Rotation - Lift - Drive");
        break;
      case 0x3C:
        Serial.println("Lift - Drive - Rotation");
        break;
      case 0xCC:
        Serial.println("Lift - Rotation - Drive");
        break;
      case 0x99:
        Serial.println("Drive speed range low");
        break;
      case 0xA5:
        Serial.println("Drive speed range medium");
        break;
      case 0xC3:
        Serial.println("Drive speed range high");
        break;
      case 0x69:
        Serial.println("Rotation speed range low");
        break;
      case 0x96:
        Serial.println("Rotation speed range medium");
        break;
      case 0x0F:
        Serial.println(" Rotation speed range high");
        break;
      default:
        Serial.println("*****   Invalid code   *****");
    }
  }
}
```