20141119 IR Remote Control - Aruino



| **Description:** |  |
| --- | --- |
| * **100% brand new and high quality** * Arduino suite by ultrathin Mini infrared wireless remote control infrared remote control and 38 KHZ infrared receiving module, the Mini ultra-thin has 20 function keys, the infrared emission furthest distance up to 8 meters, very suitable for indoor control devices of all kinds. Infrared remote control receiving module can receive standard 38 KHZ modulation signals, by means of programming the Arduino, the remote control signal decoding operation can be realized, which can produce all kinds of remote control robots and interactive work.   **Specification:**   * Battery:CR2025 Button batteries (Included) * Transmission Distance: up to 8m (depending on the surrounding environment, sensitivity of receiver etc) * Effective Angle: 60° * Static Current: 3~5uA, Dynamic Current: 3~5mA * Remote size:8.5 x 4 x 0.65cm(L x W x H) * **Note: Due to the difference between different monitors, the picture may not reflect the actual color of the item. Thank you!**   **Test Code:**   * #include <IRremote.h> * int RECV\_PIN = 11; //define input pin on Arduino * IRrecv irrecv(RECV\_PIN); * decode\_results results; * void setup() * { * Serial.begin(9600); * irrecv.enableIRIn(); // Start the receiver * } * void loop() { * if (irrecv.decode(&results)) { * Serial.println(results.value, HEX); * irrecv.resume(); // Receive the next value * } * } |  |
| **Package includes:** |  |
| * **Infrared IR Wireless Remote Control Module Kits x 1** |  |

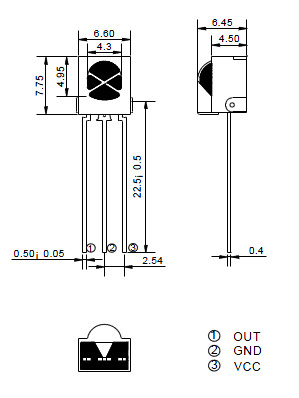
---------------------------

<http://arduino-info.wikispaces.com/IR-RemoteControl>

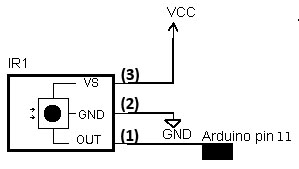
# **[IR-RemoteControl](http://arduino-info.wikispaces.com/IR-RemoteControl)**

Edit [1](http://arduino-info.wikispaces.com/page/messages/IR-RemoteControl) [43](http://arduino-info.wikispaces.com/page/history/IR-RemoteControl)[…](http://arduino-info.wikispaces.com/page/menu/IR-RemoteControl)

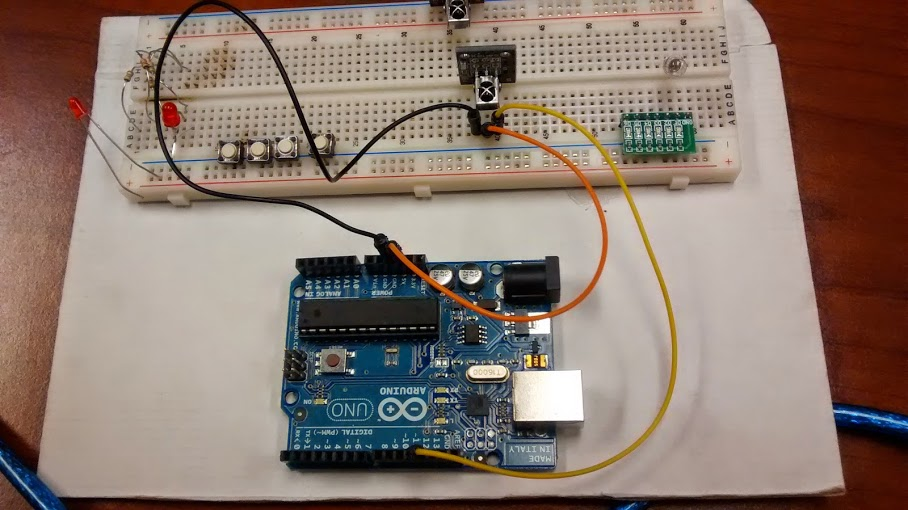
# **Infrared (IR) Remote**

****

Here's the pinout for almost every 3-pin IR Receiver:

(Below, Left) is a link to a

Jim’s wiring photo (per Leroy’s spec)



typical IR Receiver Spec Sheet:

IR-Receiver-AX-1838HS.pdf

**[IR-Receiver-AX-1838HS.pdf](http://arduino-info.wikispaces.com/file/view/IR-Receiver-AX-1838HS.pdf/264668680/IR-Receiver-AX-1838HS.pdf)**

* [Details](http://arduino-info.wikispaces.com/file/detail/IR-Receiver-AX-1838HS.pdf)
* [Download](http://arduino-info.wikispaces.com/file/view/IR-Receiver-AX-1838HS.pdf/264668680/IR-Receiver-AX-1838HS.pdf)
* 380 KB

(Above): a diagram of connecting the receiver to an Arduino. You can[get these HERE.](http://yourduino.com/sunshop2/index.php?l=product_detail&p=210)

There are many different manufacturers of IR Receivers and some have different pinouts:

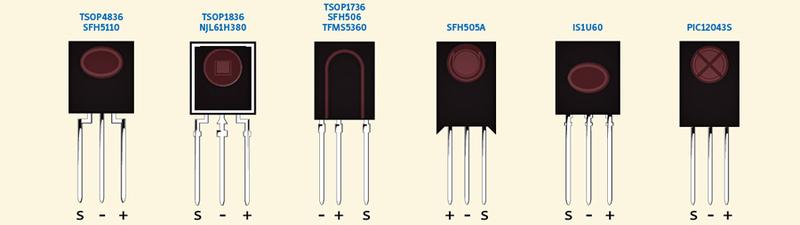
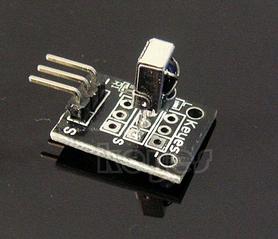


Image courtesy of Alberto Piganti. See: <http://www.pighixxx.com/>



There is also an easy-to-connect [IR Receiver Electronic Brick](http://yourduino.com/sunshop2/index.php?l=product_detail&p=368) like this (right). It can be plugged into a Sensor Shield or YourDuinoRobo1 with a 3-pin cable. This also comes with the

[IR Infrared Robot Remote Control Kit](http://yourduino.com/sunshop2/index.php?l=product_detail&p=369) which has a remote with arrow buttons for direction etc. (Scroll down for example).

[DETAILED IR REMOTE CONTROL INFORMATION](http://www.sbprojects.com/knowledge/ir/index.php) (THANKS! to San Bergmans)

## IR-REMOTE LIBRARY:

#### Note: The following library **must** be installed in your Arduino installation for this to work!

[CLICK HERE - IR REMOTE CONTROL: ARDUINO LIBRARY](https://github.com/shirriff/Arduino-IRremote)

Unzip folder into Libraries. RENAME folder IRremote.

NOTE: For Info on easier Library Installs, [SEE THIS:](http://arduino-info.wikispaces.com/Arduino-Libraries)

## TYPES OF IR REMOTE CONTROLS

There are many different IR remote controls. Some from YourDuino.com are the low-cost [IR Infrared Remote Control Kit 2](http://yourduino.com/sunshop2/index.php?l=product_detail&p=153) and also the IR Remote included as part of the [MAKER Version Electronic Brick Starter Set](http://yourduino.com/sunshop2/index.php?l=product_detail&p=364) Then, there are the typical TV and Stereo Remotes. All of these may have different encoding methods and number of physical buttons, and different codes received when a button is pressed. Below we will give example Software Sketches for a few common IR Remotes. But if you need to discover the codes received from an unknown IR Remote type, use this Sketch from the IR Remote Control Library Examples (You must first install that library - the link is above).

**NOTE!! Most handheld remotes are shipped with a small clear plastic piece in the battery compartment that must be removed to activate it. You can usually just pull it out.**

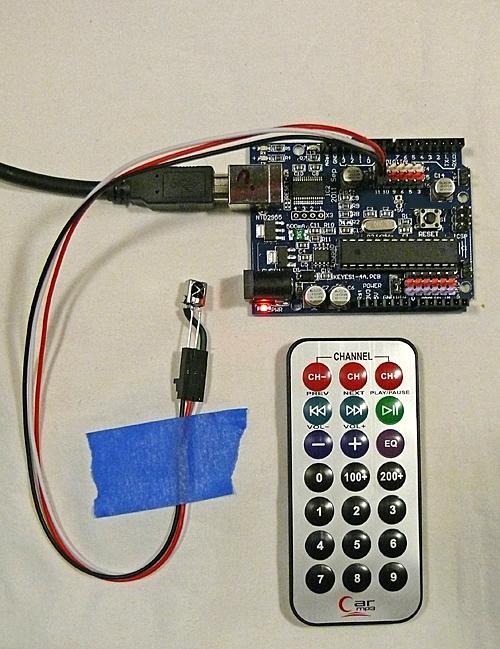
IRrecvDemo SKETCH: (Copy and paste into a blank Arduino IDE Window), Upload to your Arduino and start the Serial Monitor window:

/\*  
 \* IRremote: IRrecvDemo - demonstrates receiving IR codes with IRrecv  
 \* An IR detector/demodulator must be connected to the input RECV\_PIN.  
 \* Version 0.1 July, 2009  
 \* Copyright 2009 Ken Shirriff   
 \* <http://arcfn.com>

\* 2014 Jim McCrary - This works and is stable.  
 \*/  
  
#include <IRremote.h>  
  
int RECV\_PIN = 11;  
  
IRrecv irrecv(RECV\_PIN);  
  
decode\_results results;  
  
void **setup**()  
{  
 **Serial**.begin(9600);  
 irrecv.enableIRIn(); // Start the receiver  
}  
  
void **loop**() {  
 if (irrecv.decode(&results)) {  
 **Serial**.println(results.value, HEX);  
 irrecv.resume(); // Receive the next value  
 }  
}

### EXAMPLE: The YourDuino.com [IR Infrared Remote Control Kit 2](http://yourduino.com/sunshop2/index.php?l=product_detail&p=153)

Below is the [IR Remote Control Kit](http://yourduino.com/sunshop2/index.php?l=product_detail&p=153) connected to a YourDuinoRobo1 with a 3-pin cable. On the right is the detail of the way the IR Receiver is carefully plugged into Gnd and Vcc on the cable, and the Out pin is insulated with a piece stripped from another wire, the pins are cut off evenly, and Out is routed into the Signal (White) pin of the cable. The software below displays the button that was pressed.



Test Arduino Software Sketch for IR Infrared Remote Control Kit 2 (TESTED!!) [Other Versions below]

/\* YourDuino.com Example Software Sketch  
 IR Remote Kit Test  
 Uses YourDuino.com IR Infrared Remote Control Kit 2  
 http://arduino-direct.com/sunshop/index.php?l=product\_detail&p=153  
 based on code by Ken Shirriff - http://arcfn.com  
 Get Library at: https://github.com/shirriff/Arduino-IRremote  
 Unzip folder into Libraries. RENAME folder IRremote  
 terry@yourduino.com \*/  
  
/\*-----( Import needed libraries )-----\*/  
  
#include "IRremote.h"  
  
/\*-----( Declare Constants )-----\*/  
int receiver = 11; // pin 1 of IR receiver to Arduino digital pin 11  
  
/\*-----( Declare objects )-----\*/  
IRrecv irrecv(receiver); // create instance of 'irrecv'  
decode\_results results; // create instance of 'decode\_results'  
/\*-----( Declare Variables )-----\*/  
  
  
void **setup**() /\*----( SETUP: RUNS ONCE )----\*/  
{  
 **Serial**.begin(9600);  
 **Serial**.println("IR Receiver Raw Data + Button Decode Test");  
 irrecv.enableIRIn(); // Start the receiver  
  
}/\*--(end setup )---\*/  
  
  
void **loop**() /\*----( LOOP: RUNS CONSTANTLY )----\*/  
{  
 if (irrecv.decode(&results)) // have we received an IR signal?  
  
 {  
// Serial.println(results.value, HEX); UN Comment to see raw values  
 translateIR();   
 irrecv.resume(); // receive the next value  
 }   
}/\* --(end main loop )-- \*/  
  
/\*-----( Declare User-written Functions )-----\*/  
void translateIR() // takes action based on IR code received  
  
// describing Car MP3 IR codes   
  
{  
  
 switch(results.value)  
  
 {  
  
 case 0xFFA25D:   
 **Serial**.println(" CH- ");   
 break;  
  
 case 0xFF629D:   
 **Serial**.println(" CH ");   
 break;  
  
 case 0xFFE21D:   
 **Serial**.println(" CH+ ");   
 break;  
  
 case 0xFF22DD:   
 **Serial**.println(" PREV ");   
 break;  
  
 case 0xFF02FD:   
 **Serial**.println(" NEXT ");   
 break;  
  
 case 0xFFC23D:   
 **Serial**.println(" PLAY/PAUSE ");   
 break;  
  
 case 0xFFE01F:   
 **Serial**.println(" VOL- ");   
 break;  
  
 case 0xFFA857:   
 **Serial**.println(" VOL+ ");   
 break;  
  
 case 0xFF906F:   
 **Serial**.println(" EQ ");   
 break;  
  
 case 0xFF6897:   
 **Serial**.println(" 0 ");   
 break;  
  
 case 0xFF9867:   
 **Serial**.println(" 100+ ");   
 break;  
  
 case 0xFFB04F:   
 **Serial**.println(" 200+ ");   
 break;  
  
 case 0xFF30CF:   
 **Serial**.println(" 1 ");   
 break;  
  
 case 0xFF18E7:   
 **Serial**.println(" 2 ");   
 break;  
  
 case 0xFF7A85:   
 **Serial**.println(" 3 ");   
 break;  
  
 case 0xFF10EF:   
 **Serial**.println(" 4 ");   
 break;  
  
 case 0xFF38C7:   
 **Serial**.println(" 5 ");   
 break;  
  
 case 0xFF5AA5:   
 **Serial**.println(" 6 ");   
 break;  
  
 case 0xFF42BD:   
 **Serial**.println(" 7 ");   
 break;  
  
 case 0xFF4AB5:   
 **Serial**.println(" 8 ");   
 break;  
  
 case 0xFF52AD:   
 **Serial**.println(" 9 ");   
 break;  
  
 default:   
 **Serial**.println(" other button ");  
  
 }  
  
 delay(500);  
  
  
} //END translateIR  
  
  
  
/\* ( THE END ) \*/

[Download IRremote](https://github.com/shirriff/Arduino-IRremote)

Code corrected for this remote - Jim

ERROR PRONE - COMMENT OUT DISPLAY SECTION??

#include "IRremote.h"

/\*-----( Declare Constants )-----\*/

int receiver = 11; // pin 1 of IR receiver to Arduino digital pin 11

/\*-----( Declare objects )-----\*/

IRrecv irrecv(receiver); // create instance of 'irrecv'

decode\_results results; // create instance of 'decode\_results'

/\*-----( Declare Variables )-----\*/

void setup() /\*----( SETUP: RUNS ONCE )----\*/

{

Serial.begin(9600);

Serial.println("IR Receiver Raw Data + Button Decode Test");

irrecv.enableIRIn(); // Start the receiver

}/\*--(end setup )---\*/

void loop() /\*----( LOOP: RUNS CONSTANTLY )----\*/

{

if (irrecv.decode(&results)) // have we received an IR signal?

{

Serial.println(results.value, HEX); // UN Comment to see raw values

translateIR();

irrecv.resume(); // receive the next value

}

}/\* --(end main loop )-- \*/

/\*-----( Declare User-written Functions )-----\*/

void translateIR() // takes action based on IR code received

// describing Car MP3 IR codes

{

switch(results.value)

{

case 0xFFA25D:

Serial.println(" CHX ");

break;

case 0xFF629D:

Serial.println(" VOL+ ");

break;

case 0xFFE21D:

Serial.println(" CHX ");

break;

case 0xFF22DD:

Serial.println(" CH- ");

break;

case 0xFF02FD:

Serial.println(" OK ");

break;

case 0xFFC23D:

Serial.println(" CH+ ");

break;

case 0xFFE01F:

Serial.println(" VOLX ");

break;

case 0xFFA857:

Serial.println(" VOL- ");

break;

case 0xFF906F:

Serial.println(" EQX ");

break;

case 0xFF6897:

Serial.println(" 1 ");

break;

case 0xFF9867:

Serial.println(" 2 ");

break;

case 0xFFB04F:

Serial.println(" 3 ");

break;

case 0xFF30CF:

Serial.println(" 4 ");

break;

case 0xFF18E7:

Serial.println(" 5 ");

break;

case 0xFF7A85:

Serial.println(" 6 ");

break;

case 0xFF10EF:

Serial.println(" 7 ");

break;

case 0xFF38C7:

Serial.println(" 8 ");

break;

case 0xFF5AA5:

Serial.println(" 9 ");

break;

case 0xFF42BD:

Serial.println(" \* ");

break;

case 0xFF4AB5:

Serial.println(" 0 ");

break;

case 0xFF52AD:

Serial.println(" # ");

break;

default:

Serial.println(" other button ");

}

delay(500);

} //END translateIR

/\* ( THE END ) \*/

----------------------------------------------