

# CODIGO MANDO ARDUINO



This is for the Arduino Uno. Connect up the power and connect the Signal output to digital pin 2.

Once you have uploaded the program it will display the code of the key pressed in the Serial Monitor

```
// Arduino code for using the IR Remote Control Kit
// Please feel free to copy, edit, distribute, sell, plagiarise ... Whatever you want :)

// We are going to use the Port Registry to check the output of the IR sensor, this is
faster
#define IRpin_PIN      PIND          // Digital pin 2 is one of the Port D Input Pins
Register
#define IRpin          2              // Set up Digital Pin 2.
// For more on using the Port Registry see
http://www.arduino.cc/en/Reference/PortManipulation

uint16_t pulses[50][1];              // To store the duration of the LOW pulse
uint8_t currentPulse = 0;             // Index for pulses we're storing

void setup()
{
    Serial.begin(9600);                // Start the serial link to show what key has been
    pressed
}

void loop()
{
```

```

uint16_t highPulse, lowPulse;      // Temporary storage for timing
highPulse = 0;                     // Set
lowPulse = 0;                      // start out with no pulse length

while (IRpin_PIN && (1 << IRpin))  // Look for a HIGH pulse
{
    highPulse++;                    // Increase the counter for duration of the HIGH pulse
    delayMicroseconds(30);          // The lower this number the more frequently we check,
30 works about right
    if ((highPulse >= 4000) && (currentPulse != 0)) // Check for time out or end of data
    {
        printPulses();              // We have reached the end ( or timed out ) so print
out what we have
        currentPulse=0;              // Reset the pulse count ready for a new key press
        return;                      // Start again
    }
}

pulses[currentPulse][0] = highPulse; //Save the reading in the array

// Now we have detected and measured a HIGH pulse we look for a LOW one
while (!(IRpin_PIN && _BV(IRpin)))  // Look for a LOW pulse
{
    lowPulse++;                      // Increase the counter for duration of the LOW
pulse
    delayMicroseconds(30);           // The lower this number the more frequently we
check, 30 works about right

    if ((lowPulse >= 4000) && (currentPulse != 0)) // Check for time out or end of
data
    {
        printPulses();              // We have reached the end ( or timed out ) so
print out what we have
        currentPulse=0;              // Reset the pulse count ready for a new key
press
        return;                      // Start again
    }
}

// When we reach here we have read one HIGH - LOW set so increase our currentPulse
count and start looking for the next

```

```

        currentPulse++;
    }
    void printPulses()
    {
        // Although the IR remote sends quite a few pulses we actually only need to look at a
        few
        // to differentiate between the keys so we will just look at them.
        for (int i = 17; i < 24; i++)
        {
            // Strictly we do not need the next if / else, but it just cleans up the array so it
            just contains
            // 1s and 0s which will make it easier for you to decode

            if (pulses[i+1][0] > 50)
            {
                pulses[i+1][0] = 1;
            }
            else
            {
                pulses[i+1][0] = 0;
            }

            Serial.print(pulses[i+1][0], DEC); // Print the result to the Serial link
        }
        Serial.println("");
    }
}

```

The code you will see in the Serial Monitor is :

1010001 = CH-

0110001 = CH

1110001 = CH +

0010001 = Previous

0000001 = Next

1100001 = Play

1110000 = Vol Down

1010100 = Vol Up

1001000 = EQ

0110100 = 0

1001100 = 100 +

1011000 = 200 +

0011000 = 1

0001100 = 2

0111101 = 3

0001000 = 4

0011100 = 5

0101101 = 6

0100001 = 7

0100101 = 8

0101001 = 9

FIN