```
/* Raw IR decoder sketch!

 This sketch/program uses the Arduno and a PNA4602 to
 decode IR received. This can be used to make a IR receiver
 (by looking for a particular code)
 or transmitter (by pulsing an IR LED at ~38KHz for the
 durations detected

 Code is public domain, check out www.ladyada.net and adafruit.com
 for more tutorials!
 */

// We need to use the 'raw' pin reading methods
// because timing is very important here and the digitalRead()
// procedure is slower!
//uint8_t IRpin = 2;
// Digital pin #2 is the same as Pin D2 see
// http://arduino.cc/en/Hacking/PinMapping168 for the 'raw' pin mapping
#define IRpin_PIN      PIND
#define IRpin          2

// the maximum pulse we'll listen for - 65 milliseconds is a long time
#define MAXPULSE 65000

// what our timing resolution should be, larger is better
// as its more 'precise' - but too large and you wont get
// accurate timing
#define RESOLUTION 20

// we will store up to 100 pulse pairs (this is -a lot-)
```

```
uint16_t pulses[100][2];  // pair is high and low pulse
uint8_t currentpulse = 0; // index for pulses we're storing

void setup(void) {
  Serial.begin(9600);
  Serial.println("Ready to decode IR!");
}

void loop(void) {
  uint16_t highpulse, lowpulse;  // temporary storage timing
  highpulse = lowpulse = 0; // start out with no pulse length


//  while (digitalRead(IRpin)) { // this is too slow!
    while (IRpin_PIN & (1 << IRpin)) {
     // pin is still HIGH

     // count off another few microseconds
     highpulse++;
     delayMicroseconds(RESOLUTION);

     // If the pulse is too long, we 'timed out' - either nothing
     // was received or the code is finished, so print what
     // we've grabbed so far, and then reset
     if ((highpulse >= MAXPULSE) && (currentpulse != 0)) {
       printpulses();
       currentpulse=0;
       return;
     }
  }
  // we didn't time out so lets stash the reading
  pulses[currentpulse][0] = highpulse;

  // same as above
  while (! (IRpin_PIN & _BV(IRpin))) {
     // pin is still LOW
     lowpulse++;
     delayMicroseconds(RESOLUTION);
     if ((lowpulse >= MAXPULSE)  && (currentpulse != 0)) {
       printpulses();
       currentpulse=0;
       return;
     }
  }
  pulses[currentpulse][1] = lowpulse;

  // we read one high-low pulse successfully, continue!
  currentpulse++;
}

void printpulses(void) {
  Serial.println("\n\r\n\rReceived: \n\rOFF \tON");
  for (uint8_t i = 0; i < currentpulse; i++) {
    Serial.print(pulses[i][0] * RESOLUTION, DEC);
    Serial.print(" usec, ");
    Serial.print(pulses[i][1] * RESOLUTION, DEC);
    Serial.println(" usec");
  }

  // print it in a 'array' format
  Serial.println("int IRsignal[] = {");
  Serial.println("// ON, OFF ");
  for (uint8_t i = 0; i < currentpulse-1; i++) {
    //Serial.print("\t"); // tab
    Serial.print("pulseIR(");
    Serial.print(pulses[i][1] * RESOLUTION , DEC);
```

```
      Serial.print(");");
      Serial.println("");
      //Serial.print("\t");
      Serial.print("delayMicroseconds(");
      Serial.print(pulses[i+1][0] * RESOLUTION , DEC);
      Serial.println(");");

    }
    //Serial.print("\t"); // tab
    Serial.print("pulseIR(");
    Serial.print(pulses[currentpulse-1][1] * RESOLUTION, DEC);
    Serial.print(");");

  }
```