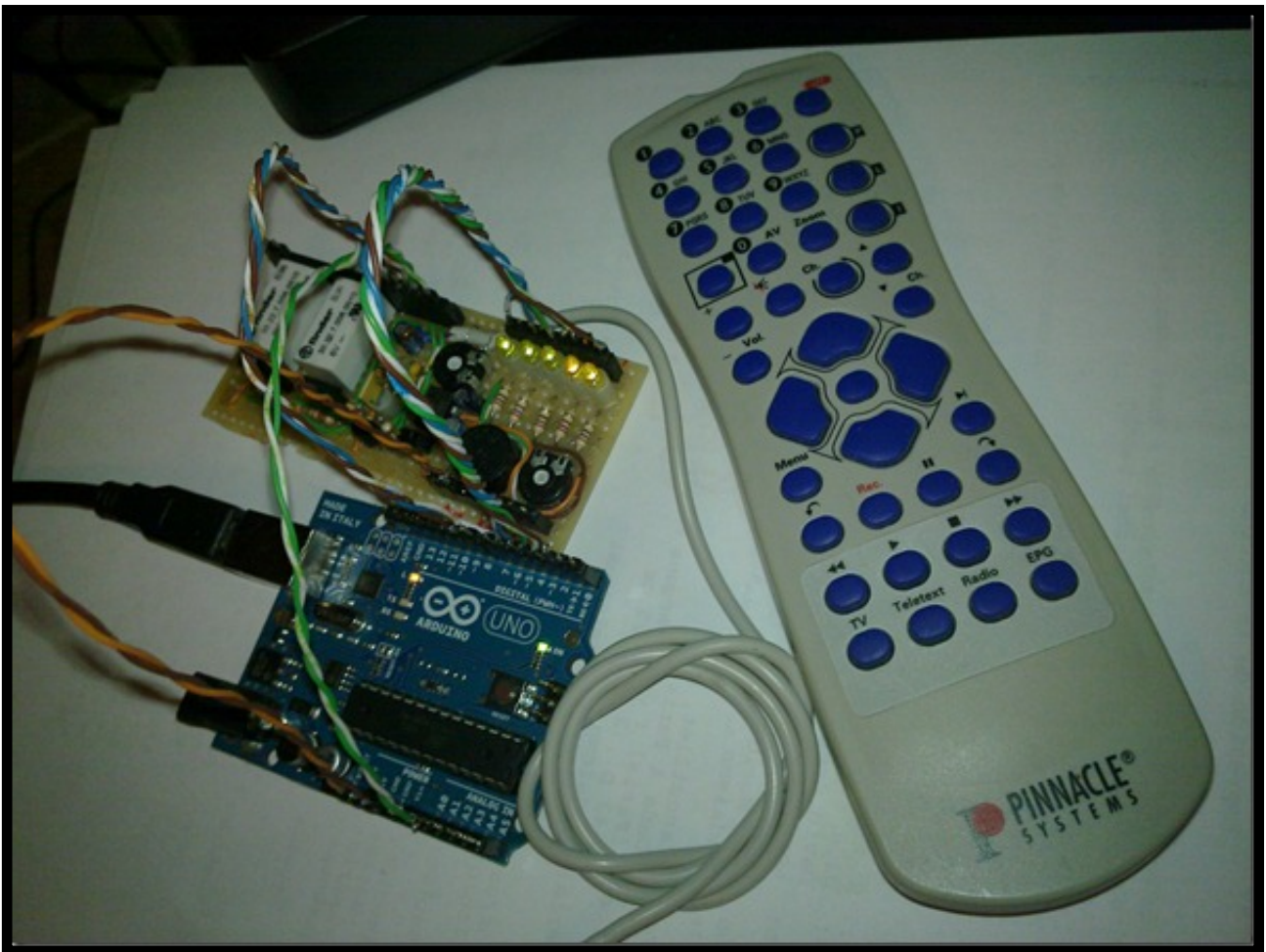


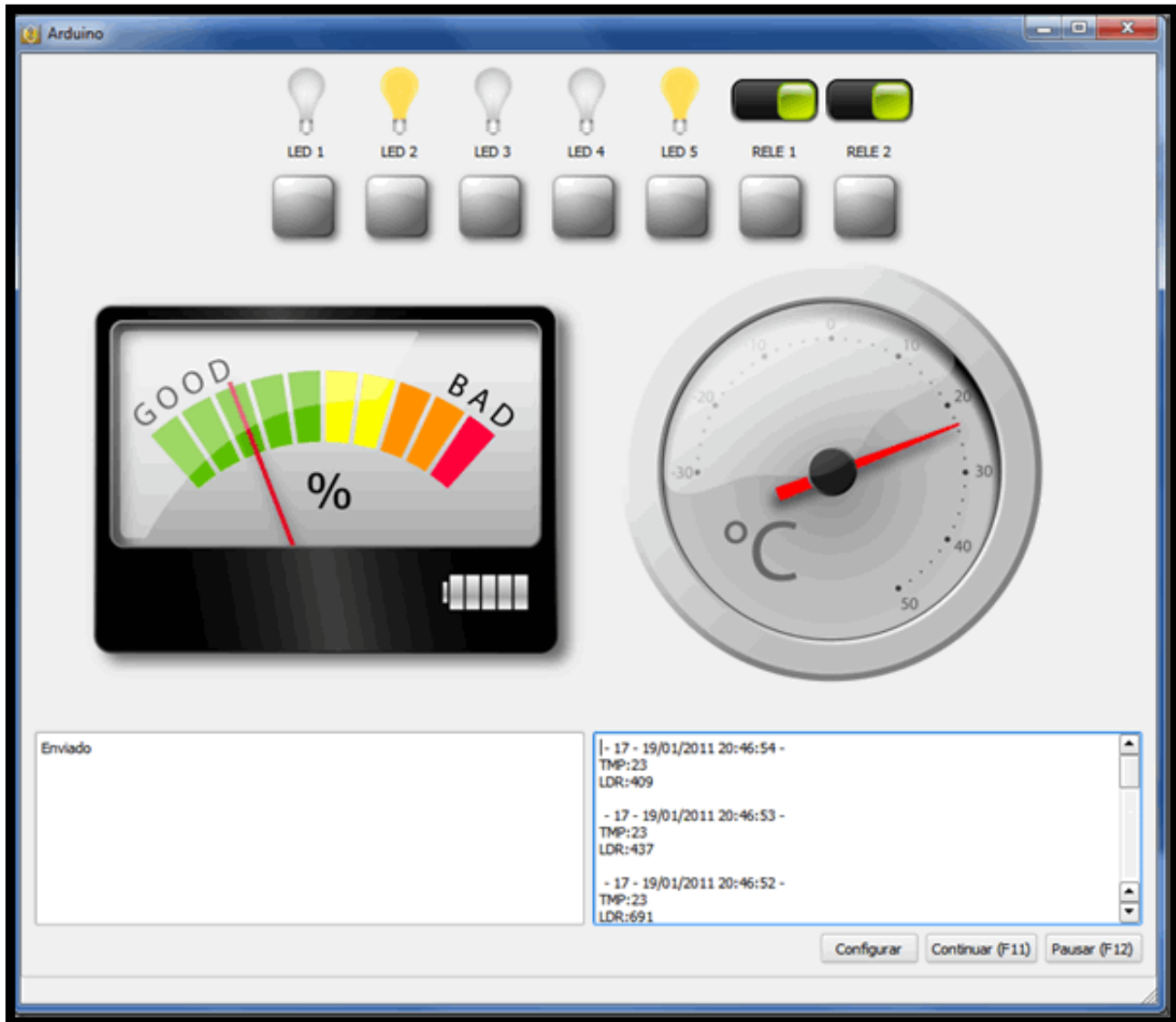
## Comunicación bidireccional Arduino y Velneo

En nuestro primer montaje con Arduino hemos decidido integrar un conjunto de hardware sencillo con Velneo. Para ello nos aprovecharemos de las comunicaciones puerto serie que Arduino y Velneo nos facilitan.

El resultado es un desarrollo sencillo y fácilmente ampliable que permite encender o apagar cualquiera de los relés desde el mando a distancia o desde el panel de control de Velneo.

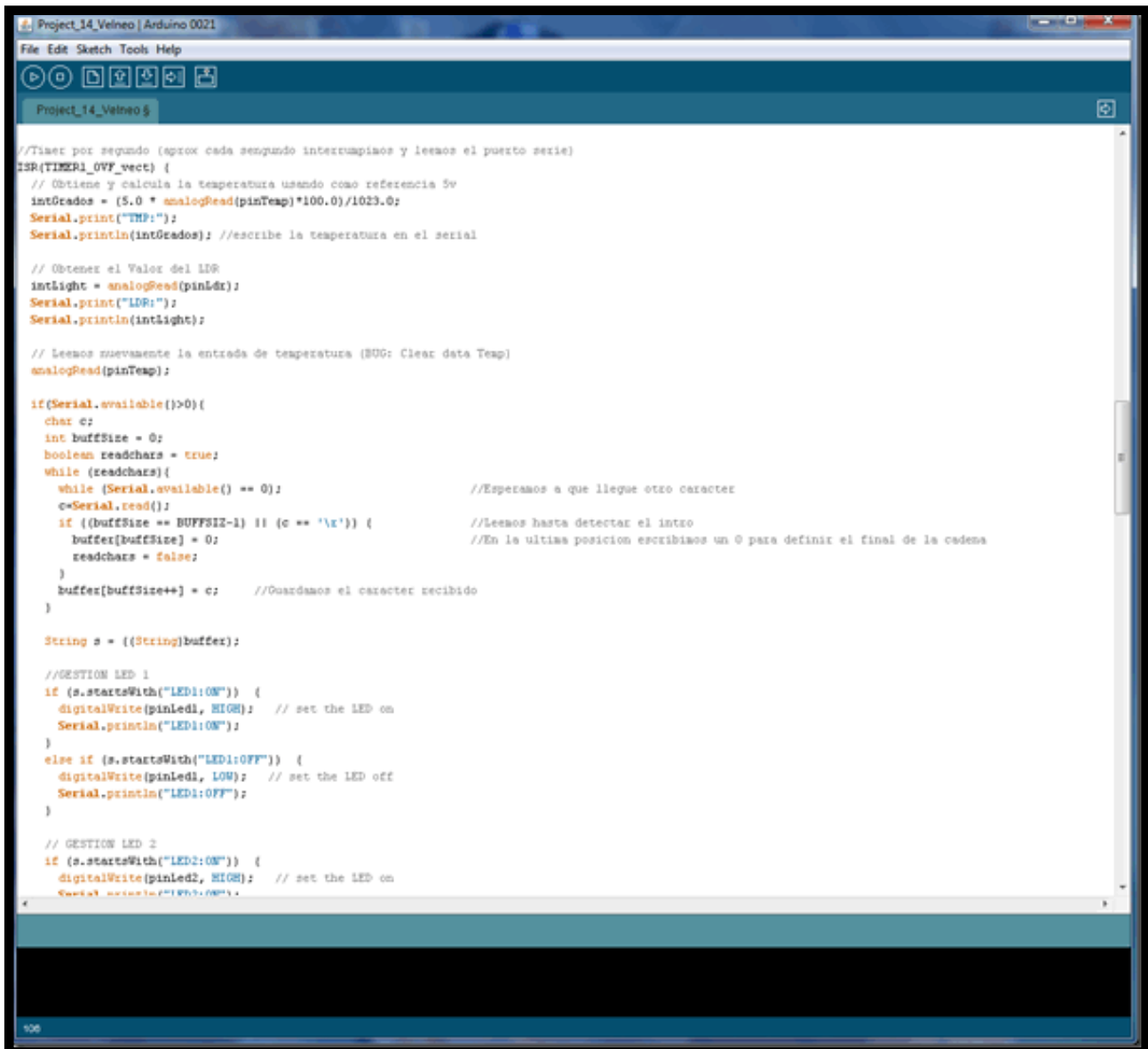


Además nos muestra el nivel de luminosidad de la habitación así como la temperatura de la misma.



El código fuente de Velneo es realmente sencillo y no requiere de ninguna habilidad especial.

En Arduino la cosa es un poco más liosa ya que tendremos que atender las lecturas de los sensores, de las pulsaciones del mando de infrarrojos y de interacción desde el panel de control de Velneo. Estas comunicaciones bidireccionales son siempre liosas. Para resolverlo hemos tenido que separar la lectura del mando a distancia (bloqueante ya que tenemos que escuchar continuamente) del resto (que es tratada mediante un perrito guardián con una interrupción aproximadamente cada segundo).



```
//Timer por segundo (aprox cada segundo interrumpimos y leemos el puerto serie)
ISR(TIMER1_OVF_vect) {
  // Obtiene y calcula la temperatura usando como referencia 5v
  intGrados = (5.0 * analogRead(pinTemp)*100.0)/1023.0;
  Serial.print("TMP:");
  Serial.println(intGrados); //escribe la temperatura en el serial

  // Obtener el Valor del LDR
  intLight = analogRead(pinLDR);
  Serial.print("LDR:");
  Serial.println(intLight);

  // Leemos nuevamente la entrada de temperatura (B00: Clear data Temp)
  analogRead(pinTemp);

  if(Serial.available()>0){
    char c;
    int buffSize = 0;
    boolean readchars = true;
    while (readchars){
      while (Serial.available() == 0); //Esperamos a que llegue otro caracter
      c=Serial.read();
      if ((buffSize == BUFFSIZE-1) || (c == '\n')) { //Leemos hasta detectar el intro
        buffer[buffSize] = 0; //En la ultima posicion escribimos un 0 para definir el final de la cadena
        readchars = false;
      }
      buffer[buffSize++] = c; //Guardamos el caracter recibido
    }

    String s = ((String)buffer);

    //GESTION LED 1
    if (s.startsWith("LED1:ON")) {
      digitalWrite(pinLed1, HIGH); // set the LED on
      Serial.println("LED1:ON");
    }
    else if (s.startsWith("LED1:OFF")) {
      digitalWrite(pinLed1, LOW); // set the LED off
      Serial.println("LED1:OFF");
    }

    // GESTION LED 2
    if (s.startsWith("LED2:ON")) {
      digitalWrite(pinLed2, HIGH); // set the LED on
      Serial.println("LED2:ON");
    }
  }
}
```

Bueno otro día os damos más detalles... ya que nos gustaría abordar este mismo montaje mediante TCP/IP.