

```
/* KY001
```

```
DS18S20 Hőmérséklet-érzékelő chip i/o
```

Mérés a DS18B20-as chipet tartalmazó hőmérsékletérzékelő modullal.

Az áramkör:

Csatlakoztassa a "Signal"-t az Arduino digitális 10-es kivezetéséhez

A szükséges könyvtár letölthető:

<https://github.com/PaulStoffregen/OneWire/archive/refs/heads/master.zip>

```
*/
```

```
// a szükséges könyvtár:
```

```
#include <OneWire.h>
```

```
OneWire ds(10); // az érzékelő a 10 kivezetésen
```

```
void setup(void) {
```

```
  // Inicializáljuk a soros portot:
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop(void) {
```

```
  byte i;      // változó a ciklusokhoz
```

```
  byte present = 0;
```

```
  byte data[12];
```

```
  byte addr[8];
```

```
  int Temp;
```

```
  if ( !ds.search(addr) ) {
```

```
    Serial.print("Nincs több cím.\n");
```

```
    ds.reset_search();
```

```
    return;
```

```
  }
```

```
  Serial.print("R=");
```

```
  for( i = 0; i < 8; i++) {
```

```
    Serial.print(addr[i], HEX);
```

```
    Serial.print(" ");
```

```
  }
```

```
  if ( OneWire::crc8( addr, 7) != addr[7] ) {
```

```
    Serial.print("A CRC nem érvényes!\n");
```

```
    return;
```

```
  }
```

```
  if ( addr[0] != 0x28 ) {
```

```
    Serial.print("Az eszköz nem a DS18S20 család eszköze.\n");
```

```
    return;
```

```
  }
```

```
  ds.reset();
```

```
  ds.select(addr);
```

```
  ds.write(0x44,1);      // az átalakítás megkezdése, a végén bekapcsolva a parazita tápellátást
```

```
  delay(1000);          // talán 750ms elég, lehet hogy nem
```

```
  // itt csinálhatunk egy ds.depower()-t, de a resetelés megoldja.
```

```
  present = ds.reset();
```

```
  ds.select(addr);
```

```
  ds.write(0xBE); // Scratchpad olvasása
```

```
  Serial.print("P=");
```

```
  Serial.print(present,HEX);
```

```
  Serial.print(" ");
```

```
  for ( i = 0; i < 9; i++) { // 9 bájtra van szükségünk
```

```

data[i] = ds.read();
Serial.print(data[i], HEX);
Serial.print(" ");
}
Temp = (data[1] << 8) + data[0]; // vegyen ki a két bájtot a hőmérsékletre vonatkozó válaszból
Temp = Temp >> 4; // ossza el 16-tal, hogy tiszta Celsius-leolvasást kapjon
// a következő sor a Fahrenheit-konverzió
Temp = Temp * 1.8 + 32; // kommentelje ezt a sort, hogy celsius legyen
Serial.print("T="); //kiírja a hőmérsékletet a soros portra
Serial.print(Temp);
Serial.print(" ");
Serial.print(" CRC=");
Serial.print( OneWire::crc8( data, 8), HEX);
Serial.println();
}

```

(A fenti az általam módosított vázlat. Íme az eredeti, amely azonban hiányos:)

Connect the signal to Arduino digital 10, and open serial port monitor:

```

#include
/* DS18S20 Temperature chip i/o
*/
OneWire ds(10); // on pin 10
void setup(void) {
// initialize inputs/outputs
// start serial port
Serial.begin(9600);
}
void loop(void) {
byte i;
byte present = 0;
byte data[12];
byte addr[8];
int Temp;
if ( !ds.search(addr) ) {
//Serial.print("No more addresses.n");
ds.reset_search();
return;
}
Serial.print("R="); //R=28 Not sure what this is
for( i = 0; i < 8; i++) {
Serial.print(addr[i], HEX);
Serial.print(" ");
}
if ( OneWire::crc8( addr, 7) != addr[7] ) {
Serial.print("CRC is not valid!n");
return;
}
if ( addr[0] != 0x28 ) {
Serial.print("Device is not a DS18S20 family device.n");
return;
}
ds.reset();
ds.select(addr);
ds.write(0x44,1); // start conversion, with parasite power on at the end
delay(1000); // maybe 750ms is enough, maybe not
// we might do a ds.depower() here, but the reset will take care of it.

```

```
present = ds.reset();
ds.select(addr);
ds.write(0xBE); // Read Scratchpad
Serial.print("P=");
Serial.print(present,HEX);
Serial.print(" ");
for ( i = 0; i < 9; i++) { // we need 9 bytes
data[i] = ds.read();
Serial.print(data[i], HEX);
Serial.print(" ");
}
Temp=(data[1]<<8)+data[0]; //take the two bytes from the response relating to
temperature
Temp=Temp>>4; //divide by 16 to get pure celcius readout
//next line is Fahrenheit conversion
Temp=Temp*1.8+32; // comment this line out to get celcius
Serial.print("T="); //output the temperature to serial port
Serial.print(Temp);
Serial.print(" ");
Serial.print(" CRC=");
Serial.print( OneWire::crc8( data, 8), HEX);
Serial.println();
}
```