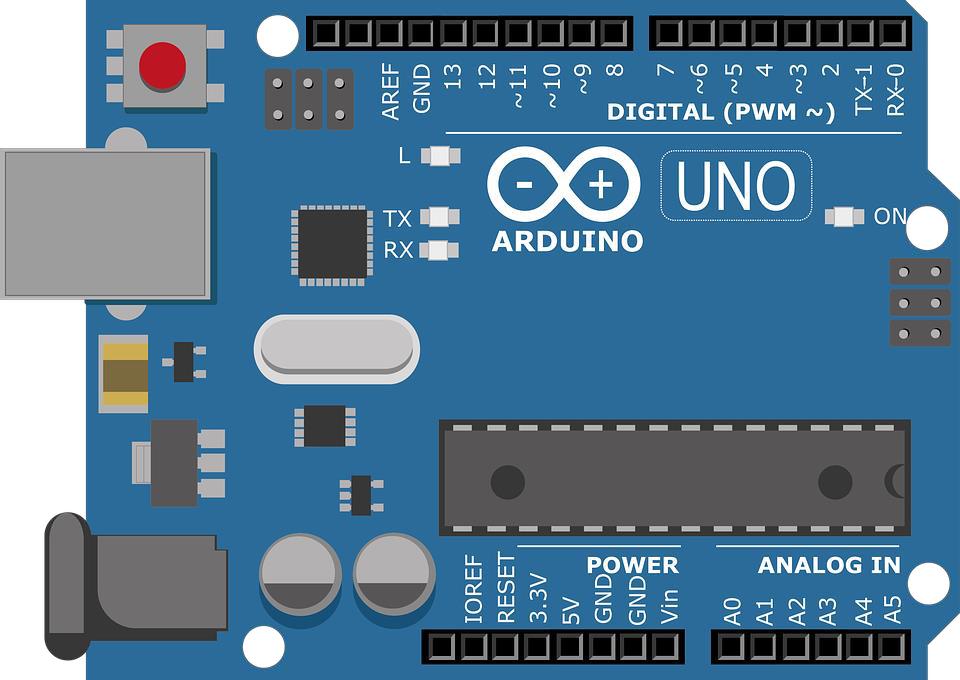
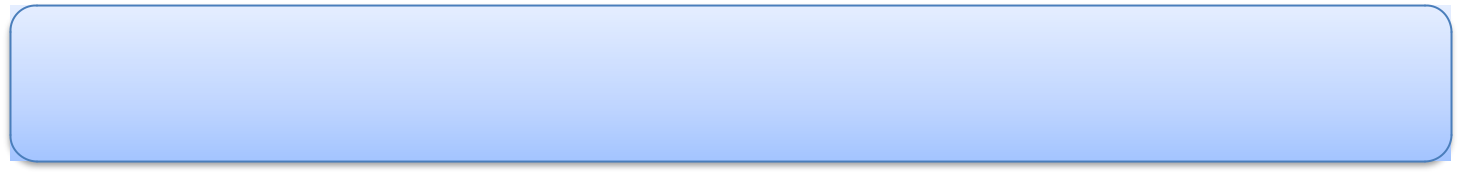
Arduino



Arduino is an open-source electronics platform

 Arduino



Сіз қарапайым Arduino-мен таныссыз деп болжануда (егер жоқ болса, Arduino бағдарламасын үйрену өте қарапайым!).

Егер сіз өз біліміңізді жаңарту керек деп санасаңыз, онда төменде берілген жаттығуларды орындаған жөн (егер олай

болмаса, оларды өткізіп жіберуге болады).

Ардуиноны үйрену үшін келесі негізгі әрекеттерді орындаңыз

Arduino жаттығулары:

• Жарықдиодты жыпылықтаңыз

• Светодиодты қосқыш / басу түймесін пайдаланып қосыңыз

• Жарықдиодтың қарқындылығын төмендету / жоғарылату үшін потенциометрді қолданыңыз («жарық диммер»)

• TMP36 көмегімен температура мәндерін оқыңыз

• NTC Thermistor көмегімен температура мәндерін оқыңыз



• Arduino - бұл электроникамен тәжірибе жасауға және бағдарламалауды көңілді әрі интуитивті етуге арналған ашық көзді физикалық есептеу платформасы.

• Arduino өзінің ерекше, жеңілдетілген бағдарламалау тіліне ие және көптеген алдын ала емтихандар мен оқулықтар бар.

• Arduino көмегімен сіз қозғалтқыштар, температура датчиктері және т.с.с. шағын көлемді датчиктер мен қозғағыштарды оңай зерттей аласыз.

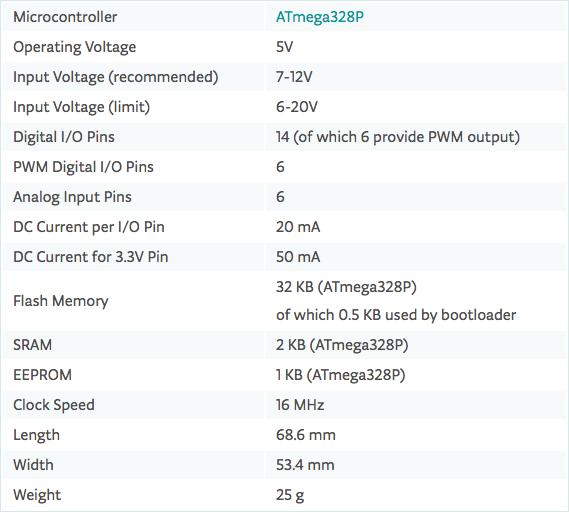
• Arduino-мен мүмкіндіктер шексіз.

[http://www.arduino.cc](http://www.arduino.cc/)

Arduino UNO

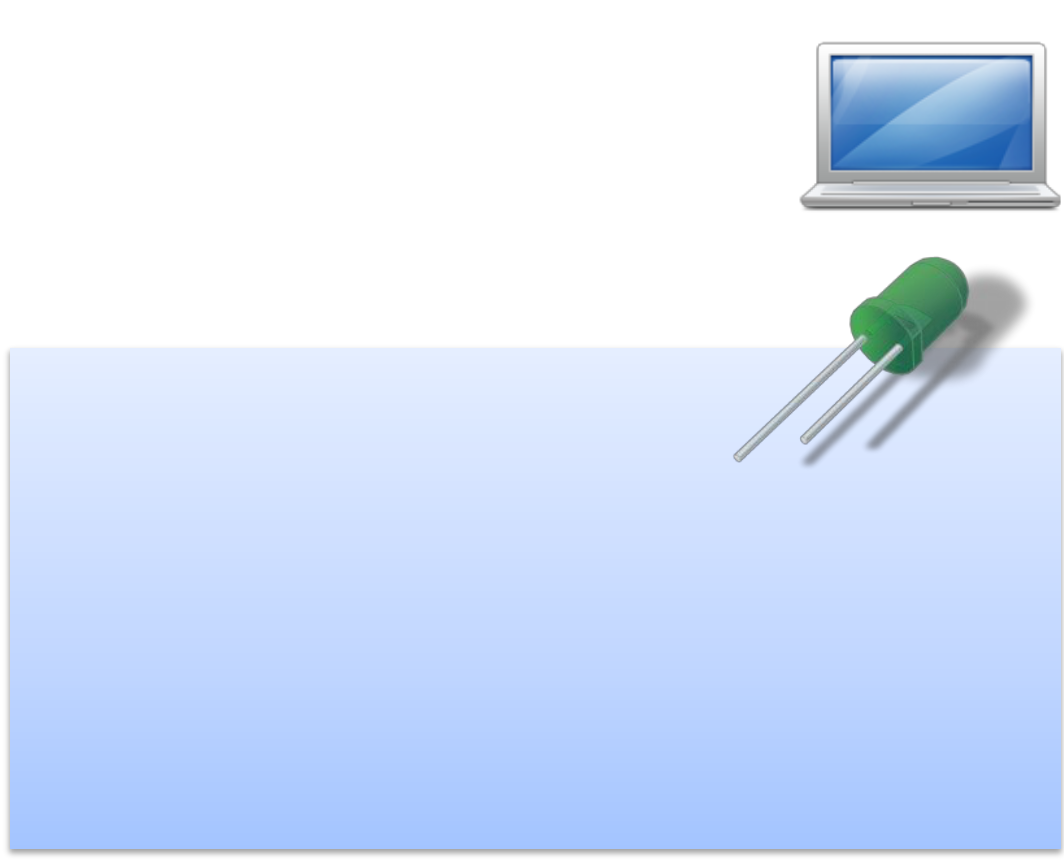


<https://www.arduino.cc/en/Guide/HomePage>

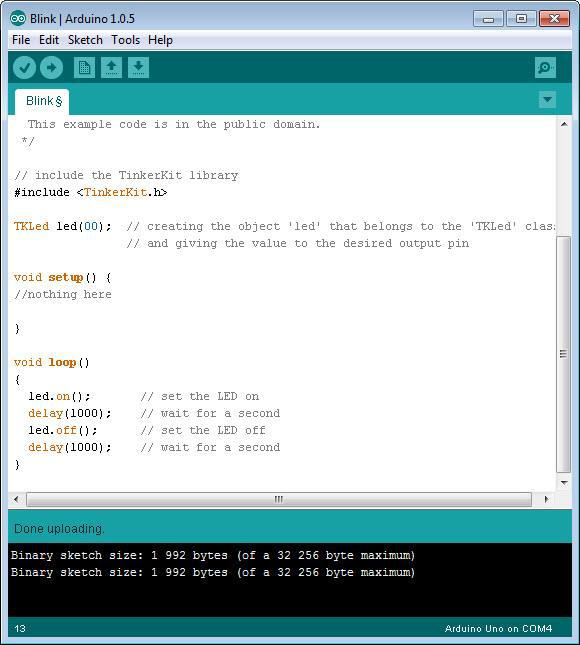


[http://www.arduino.cc](http://www.arduino.cc/)

Pin Overview: <http://pighixxx.com/unov3pdf.pdf>



|  |  |  |  |
| --- | --- | --- | --- |
| Arduino | Software Бағдарламалық жасақтама | | Software |
| Arduino Sketch IDE | Programming with Arduino is simple and intuitive! | |  |
|  | Example: |  |
|  |  |  |
|  |  |  |  |
|  |  | void setup() |  |
|  |  | { |  |
|  |  | pinMode(8, OUTPUT); |  |
|  |  | } |  |
|  |  | void loop() |  |
|  |  | { |  |
|  |  | **digitalWrite(8, HIGH);** | // Turn on the LED |
|  |  | delay(1000); | // Wait for one second |
|  |  | **digitalWrite(8, LOW);** | // Turn off the LED |
|  |  | delay(1000); | // Wait for one second |
|  |  | } |  |
|  |  |  |  |

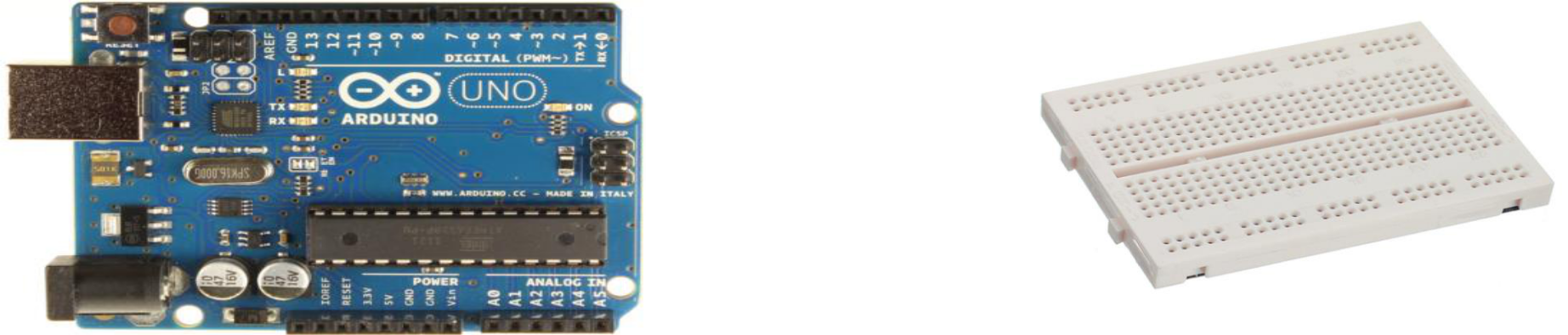


This program makes a LED blink

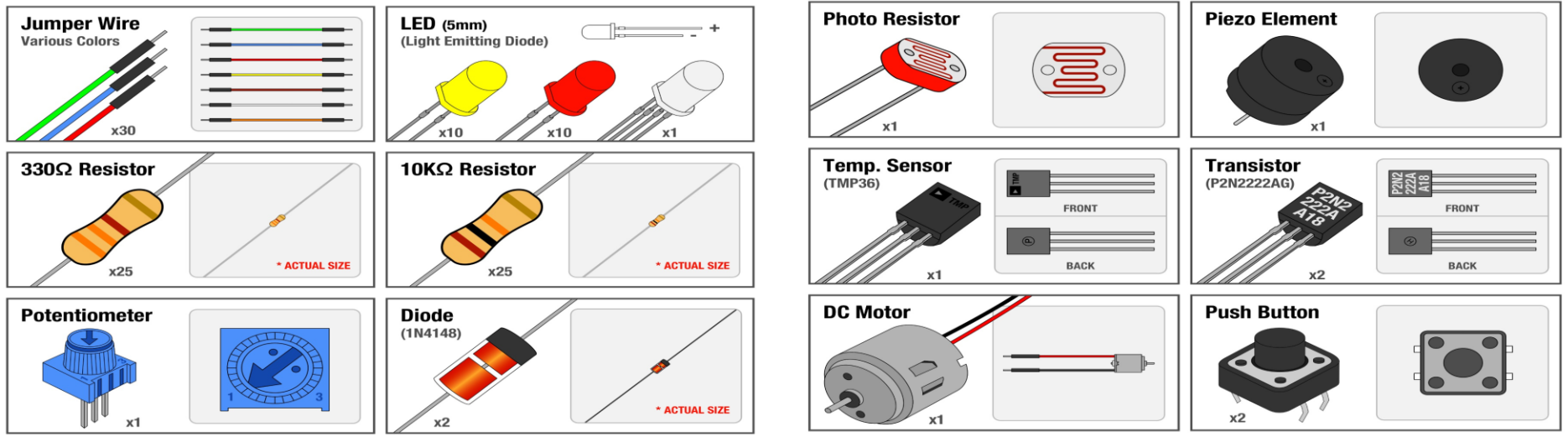
The syntax is similiar to C programming

Software Installation: <http://arduino.cc/en/Main/Software>

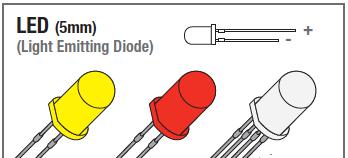
Arduino Uno Board Arduino Basics Breadboard

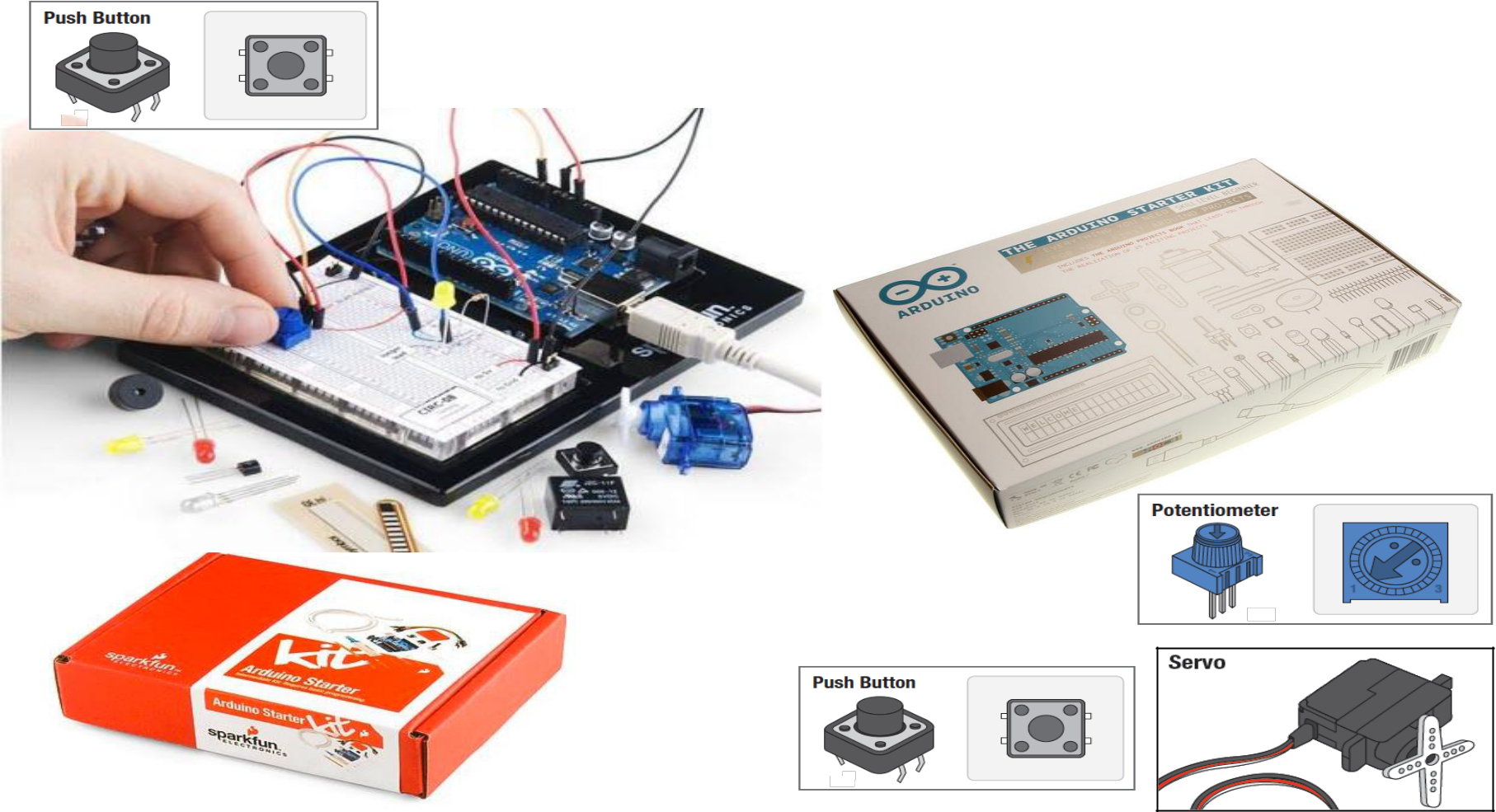


Sensors and Actuators, etc.



**26**

The Arduino Kit 



Arduino Uno Board

Small-size Sensors and Actuators

Getting Started with Arduino: <http://arduino.cc/en/Guide/HomePage>

The Arduino Kit

* Arduino Home Page: [http://arduino.cc](http://arduino.cc/)
* The Arduino Starter Kit: <http://arduino.cc/en/Main/ArduinoStarterKit>
* Starter Kit Videos:

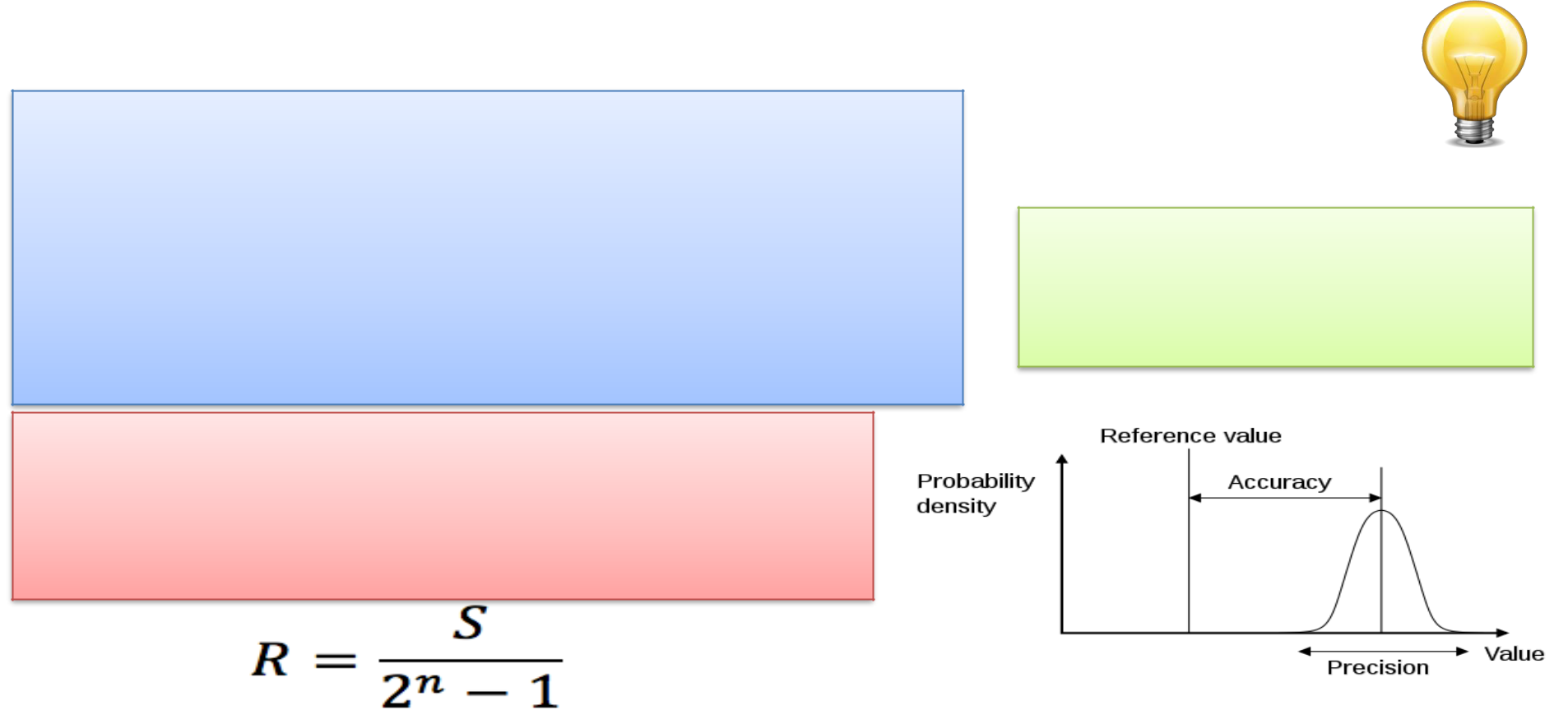
[https://www.youtube.com/playlist?feature=edit\_ok&list=PLT6](https://www.youtube.com/playlist?feature=edit_ok&list=PLT6rF_I5kknPf2qlVFlvH47qHvqvzkknd) [rF\_I5kknPf2qlVFlvH47qHvqvzkknd](https://www.youtube.com/playlist?feature=edit_ok&list=PLT6rF_I5kknPf2qlVFlvH47qHvqvzkknd)



|  |  |
| --- | --- |
| Датчиктер мен жетектер | Theory |
|  |

• Сенсор - бұл физикалық шаманы өлшейтін және оны бақылаушы немесе (бүгінде электронды) құрал оқи алатын сигналға айналдыратын түрлендіргіш.

• Жетек - бұл механизмді немесе жүйені басқаруға немесе басқаруға арналған қозғалтқыш түрі. Ол энергия көзі, әдетте электр тогы, сұйықтықтың гидравликалық қысымы немесе пневматикалық қысыммен жұмыс істейді және сол энергияны қозғалысқа айналдырады. Орындаушы - бұл басқару жүйесінің қоршаған ортаға әсер ету механизмі.

. 

Cенсорлар

Калибрлеу: Өлшеуді салыстыру. Бір құрылғы арқылы жасалған немесеорнатылған белгілі бір шамада немесе дәлдікте және екінші құрылғыда мүмкіндігінше ұқсас түрде жасалған басқа өлшеуде.

Белгілі немесе тағайындалған дұрыстығы бар құрылғы стандартты деаталады. Екінші құрылғы - бұл сыналатын қондырғы, сыналатын құрал немесе калибрленетін құрылғыға арналған кез-келген басқаатаулар

Theory

Дәлдік: өлшенген мәннің нақты / нақты мәні қаншалықты жақын, мысалы, ± 0,1%

Ажыратымдылық: ол өлшейтін мөлшерде ең аз өзгерісті

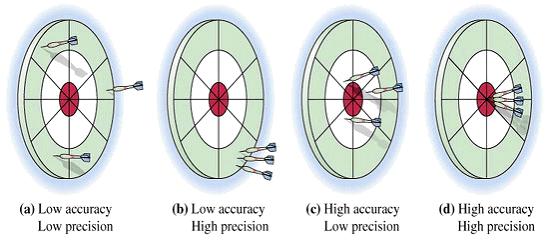
анықтай алады.

Келесі формуланы қолдануға болады (мұндағы

S - өлшеу аралығы, мысалы, 0-100de.C):

Тапсырмада сіз осы параметрлермен жұмыс істеуіңіз керек. Сіз осы параметрлер туралы ақпаратты Деректерден таба аласыз құрылғыға арналған парақ

Олшеу және датчиктер



Measurement Fundamentals: <http://www.ni.com/white-paper/4523/en/>

Sensor Fundamentals: <http://www.ni.com/white-paper/4045/en/>

Sensor Terminology: <http://www.ni.com/white-paper/14860/en/>



**Кернеуге негізделген сенсорлар**

TMP36 деректер кестесіне сәйкес шығыс кернеуінің қатынасы

нақты температураға дейін осы теңдеу қолданылады: TMP36

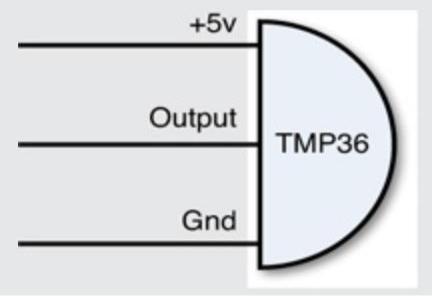
y[°C] = (1/10)\*x[mv]-50

Мұнда кернеу мәні милливольтпен көрсетілген.

Алайда, сіз бұл теңдеуді қолданар алдында, деп бүтін мәнді түрлендіруіңіз керек

analogRead функциясы милливольт мәніне оралады.

Сіз білесіз бе, 5000 мВ (5 В) мәні үшін analogRead функциясы 1024 мүмкін мәнді береді:

voltage = (5000 / 1024) \* output

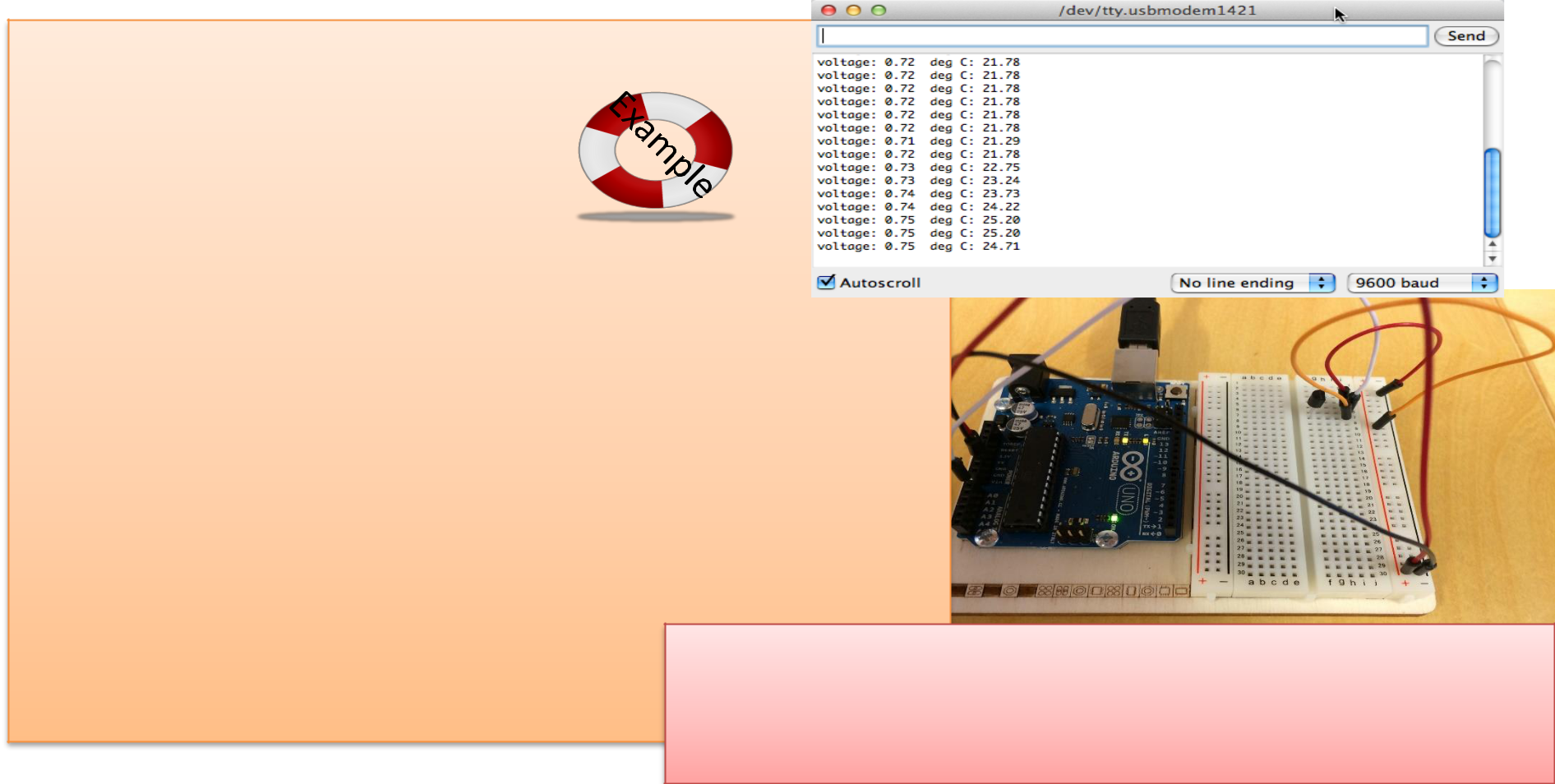
Where mV

output = analogRead(aichannel)

0-1023 A0-A5

Цифрлық түрлендіргішке 10 биттік аналог

**TMP36 температура сенсорының мысалы**



* We'll use analog input 0 to read Temperature Data const int temperaturePin = 0;

void **setup()**

{

Serial.begin(9600);

}

void **loop()**

{

float voltage, degreesC, degreesF;

**voltage = getVoltage(temperaturePin);**

* Now we'll convert the voltage to degrees Celsius.
* This formula comes from the temperature sensor datasheet:

**degreesC = (voltage - 0.5) \* 100.0;**

* Send data from the Arduino to the serial monitor window Serial.print("voltage: ");

Serial.print(voltage); Serial.print(" deg C: "); Serial.println(degreesC);

delay(1000); // repeat once per second (change as you wish!)

}

float **getVoltage(int pin)**

{

**return (analogRead(pin) \* 0.004882814);**

* This equation converts the 0 to 1023 value that analogRead()
* returns, into a 0.0 to 5.0 value that is the true voltage
* being read at that pin.

}

Serial Monitor

**Қарсылыққа негізделген сенсорлар**

Theory

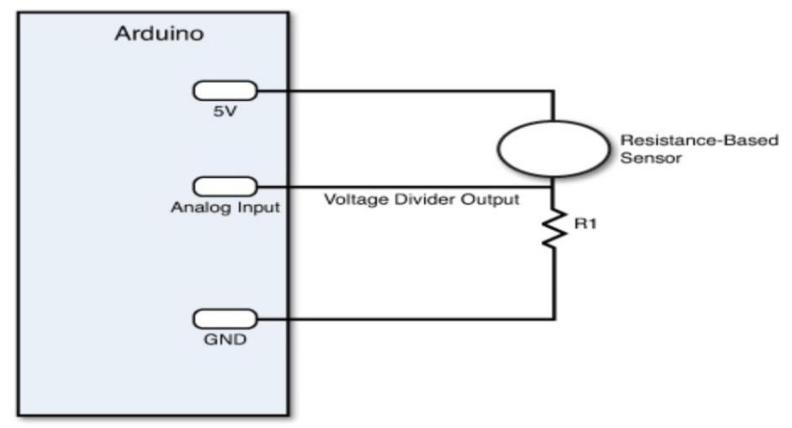


Қарсылық датчиктеріндегі проблема Arduino аналогында

интерфейстер қарсылықтың өзгеруін тікелей анықтай алмайды. Термистор Бұл үшін қосымша электронды компоненттер қажет. Ең оңай жолы

қарсылықтың өзгеруін анықтау - бұл өзгерісті кернеудің өзгеруіне айналдыру. Сіз мұны төменде көрсетілгендей кернеу бөлгішті қолданып жасайсыз.



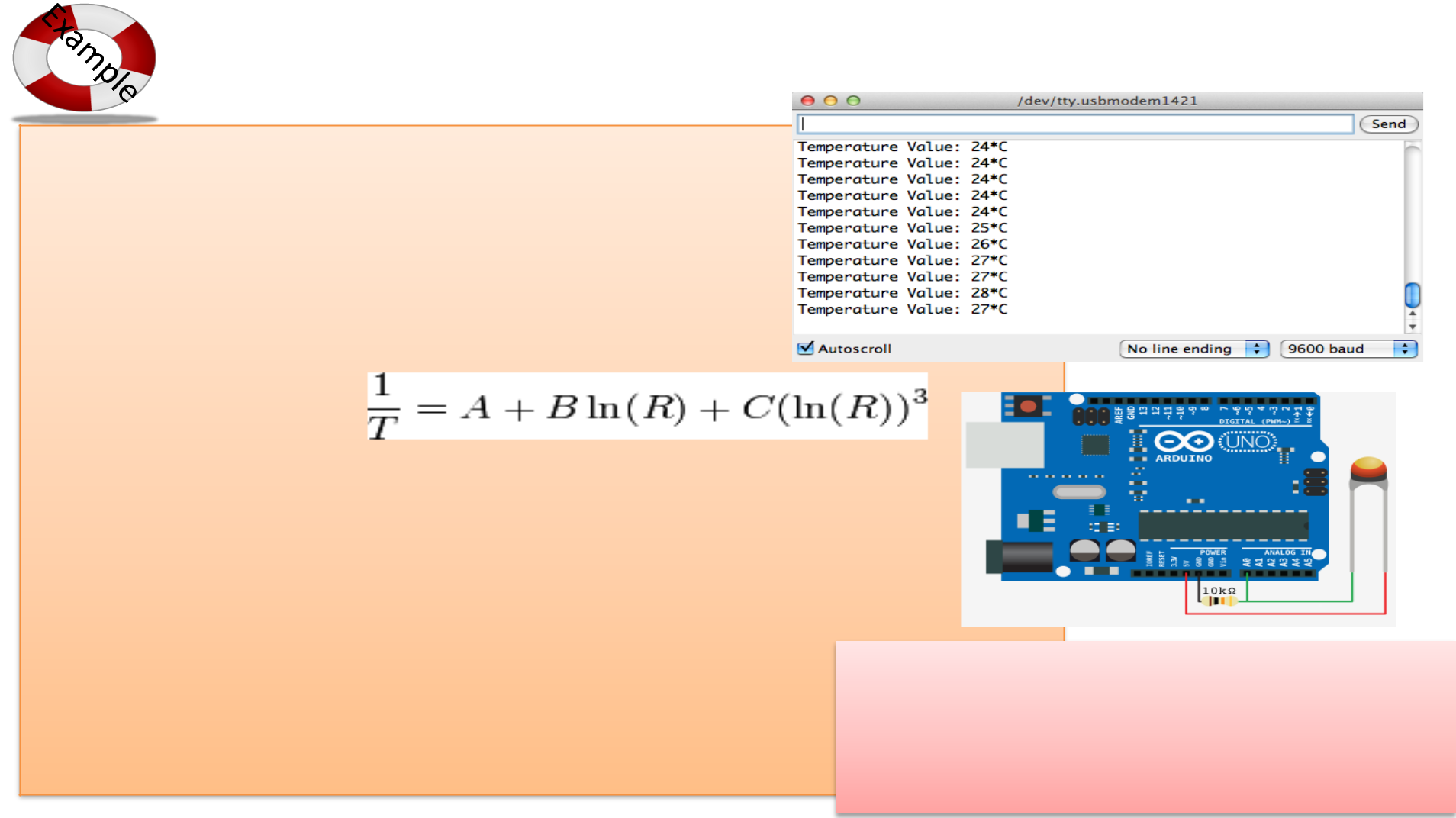


Мысалы, температураны табу үшін Штейнхарт-Харт теңдеуін қолдануға болады:



Қуат көзінің шығуын тұрақты ұстап тұру арқылы, датчиктің кедергісі өзгерген сайын, кернеуді бөлгіш тізбегі өзгереді, ал шығыс кернеуі өзгереді. R1 резисторы үшін сізге қажет резистордың мөлшері сенсор тудыратын қарсылық ауқымына және шығыс кернеуінің өзгеруін қалай сезінетіндігіңізге байланысты.

Әдетте, 1K мен 10K ом арасындағы мән Arduino аналогтық кіріс интерфейсінде анықтауға болатын шығыс кернеуді жасау үшін өте жақсы жұмыс істейді.



NTC термисторының мысалы

* Read Temerature Values from NTC Thermistor const int temperaturePin = 0;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| void setup() | |  |  |  |  |
| { |  |  |  |  |  |
| Serial.begin(9600); | |  |  |  |  |
| } |  |  |  |  | Serial Monitor |
| void loop() | |  |  |  |
|  |  |  |  |
| { |  |  |  |  |  |
| int temperature = getTemp(); | |  |  |  |  |
| Serial.print("Temperature Value: "); | | Steinhart-Hart Equation: |  |  |  |
| Serial.print(temperature); | |  |  |  |
| Serial.println("\*C"); | |  |  |  |
|  |  |  |  |
| delay(1000); |  |  |  |  |  |
| } |  |  |  |  |  |
| double getTemp() |  |  |  |  |  |
| { |  |  |  |  |  |
| // Inputs ADC Value from Thermistor and outputs Temperature in Celsius | | |  |  |  |
| int RawADC = analogRead(temperaturePin); | |  |  |  |  |
| long Resistance; | |  |  |  |  |
| double Temp; | |  |  |  |  |
| // Assuming a 10k Thermistor. Calculation is actually: Resistance = (1024/ADC) | | |  |  |  |
| Resistance=((10240000/RawADC) - 10000); | |  |  |  |  |
| // Utilizes the Steinhart-Hart Thermistor Equation: | | |  |  |  |
| // Temperature in Kelvin = 1 / {A + B[ln(R)] + C[ln(R)]^3} | | |  |  |  |
|  |  |  |  | | |
|  |  |  |  |  |  |

SPI Bus

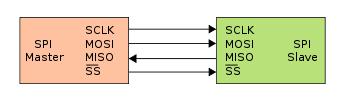
• Сериялық перифериялық интерфейс (SPI) - бұл бір немесе бірнеше перифериялық құрылғылармен қысқа қашықтықта жылдам байланыс орнатуға арналған микроконтроллерлер қолданатын синхронды сериялық деректер хаттамасы.

• SPI қосылымында әрдайым перифериялық құрылғыларды басқаратын бір негізгі құрылғы болады (әдетте микроконтроллер).

• SPI құрылғылары толық дуплексті режимде бір мастермен мастер-құл архитектурасын қолдана отырып байланысады.

• Интерфейсті Motorola әзірледі және іс жүзінде стандартқа айналды.

• Әдеттегі қосымшаларға датчиктер, Secure Digital карталары және сұйық кристалды дисплейлер (LCD) кіреді.



SCLK : Serial Clock (output from master)

MOSI : Master Output, Slave Input (output from master)

MISO : Master Input, Slave Output (output from slave)

|  |  |
| --- | --- |
| SS (or SC) : Slave Select (active low, output from master) | <http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus> |
|  |

<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

I2C Bus

• I²C (Inter-Integrated Circuit) - бұл көп шеберлі, көп құлды, бір жақты, сериялық компьютерлік шина

• Әдетте бұл процессорлар мен микроконтроллерлерге төменгі жылдамдықты перифериялық ИК-терді бекіту үшін қолданылады.

• I²C әдетте I2C деп жазылады (I-two-C деп оқылады)

• I²C шинасын 1982 жылы Philips Semiconductor компаниясы жасаған.

• I²C протоколы барлық периферияны микроконтроллерге қосу үшін тек 2 сымды қажет етеді.

<https://learn.sparkfun.com/tutorials/i2c>

<http://en.wikipedia.org/wiki/I2C>

Fritzing

* • Fritzing - бұл электронды жабдықты жобалауға және схемалар құруға арналған ашық бастапқы бағдарлама
* • Құжаттау маңызды!
* • Кәсіби электр схемаларын құру өте маңызды, сондықтан сызбалар үшін Fritzing сияқты құралды қолдануды ойластырған жөн.