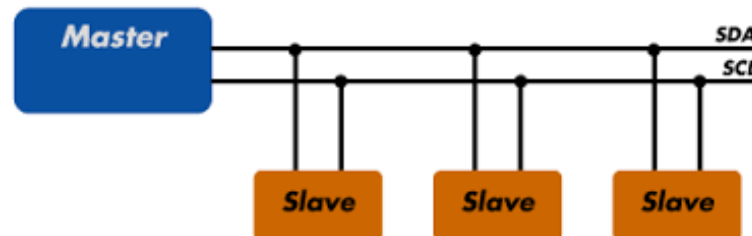


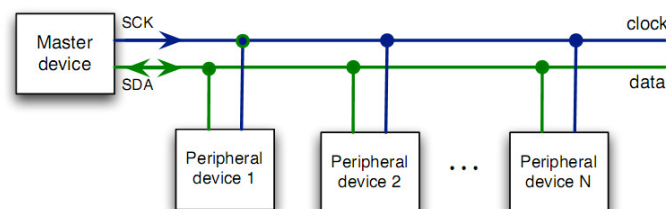
5. Inter Integrate Circuit Bus (I²C)



I²C , “Two Wire”

Synchronous serial bus with shared a data line

a little network for your gadgets



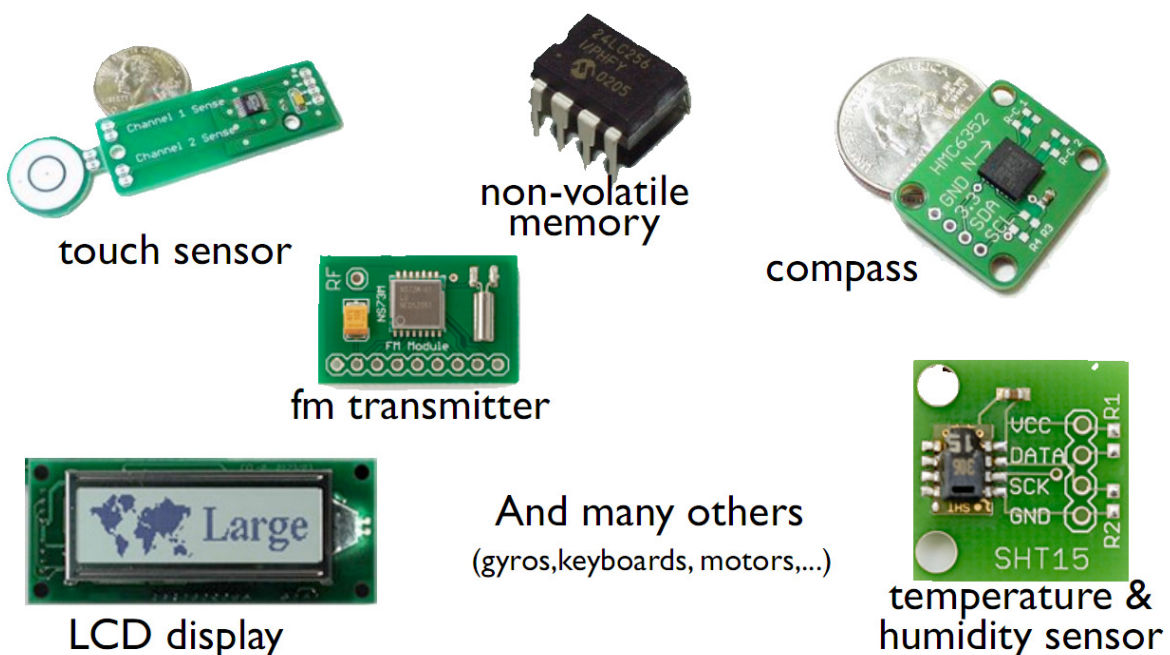
- *Up to 127 devices on one bus*
- *Up to 1Mbps data rate*
- *Really simple protocol (compared to USB,Ethernet,etc)*
- *Most microcontrollers have it built-in*

I²C , “Two Wire”

- I²C Bus ย่อมาจาก Inter Integrate Circuit Bus เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ ติดต่อสื่อสารระหว่าง ไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก
- ถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ serial data (SDA) และสาย serial clock (SCL) ซึ่งสามารถ เชื่อมต่ออุปกรณ์ จำนวนหลายๆ ตัว เข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

3

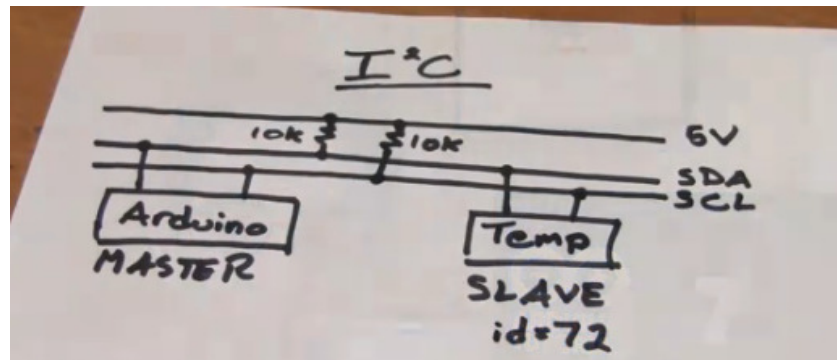
Many I²C devices



4

I²C on Arduino

- I²C built-in on Arduino's ATmega168 chip
- Use "Wire" library to access it
- Analog In 4 is SDA signal (A4 = SDA)
- Analog In 5 is SCL signal (A5 = SCL)



5

Wire library Functions

- `Wire.begin()` *join i²c bus (master)*
- `Wire.begin(4)` *join i²c bus with **address #4** (Slave)*
- `Wire.beginTransmission(112)` *transmit to **device #112***
- `Wire.endTransmission()` *stop transmitting*
- `Wire.requestFrom(112, 2)` *request **2 bytes** from slave **device #112***
- `Wire.available()` *Returns the number of bytes available for reading*
- `Wire.read()` *receive a byte*
- `Wire.write("x is ")` *sends five byte*
- `Wire.write(x)` *sends one byte*

6

Arduino “Wire Library”

Master Writing data to Slave

Load wire Library	<code>#include <Wire.h></code>
	<code>void setup()</code>
	<code>{</code>
Join i2c	<code>Wire.begin();</code> <i>// join i2c bus (master)</i>
	<code>}</code>
	<code>byte x = 0;</code>
	<code>void loop()</code>
	<code>{</code>
Start sending	<code>Wire.beginTransmission(4);</code> <i>// transmit to device #4</i>
Send data	<code>Wire.write("x is ");</code> <i>// sends five bytes</i>
	<code>Wire.write(x);</code> <i>// sends one byte</i>
Stop sending	<code>Wire.endTransmission();</code> <i>// stop transmitting</i>
	<code> </code>
	<code>x++;</code>
	<code>delay(500);</code>
	<code>}</code>

7

Arduino “Wire Library”

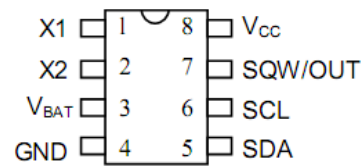
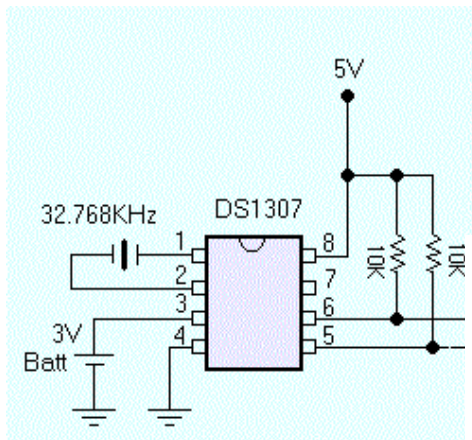
Master Reading data from Slave

	<code>#include <Wire.h></code>
	<code>void setup()</code>
	<code>{</code>
Join i2c	<code>Wire.begin();</code> <i>// join i2c bus (master)</i>
	<code>Serial.begin(9600);</code> <i>// start serial for output</i>
	<code>}</code>
	<code>void loop()</code>
	<code>{</code>
Request data	<code>Wire.requestFrom(2, 6);</code> <i>//request 6 bytes from device #2</i>
	<code>while(Wire.available())</code> <i>//slave may send less than requested</i>
Get data	<code>{</code>
	<code> char c = Wire.read();</code> <i>// receive a byte as character</i>
	<code> Serial.print(c);</code> <i>// print the character</i>
	<code>}</code>
	<code> </code>
	<code>delay(500);</code>
	<code>}</code>

8

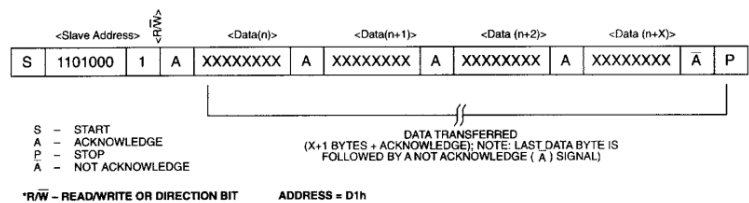
การทดลอง Temperature Sensor DS1621

- ตัวอย่างตามรูป



DS1307 8-Pin DIP (300-mil)

รายละเอียดขา

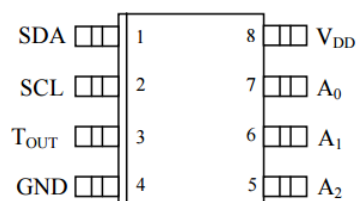
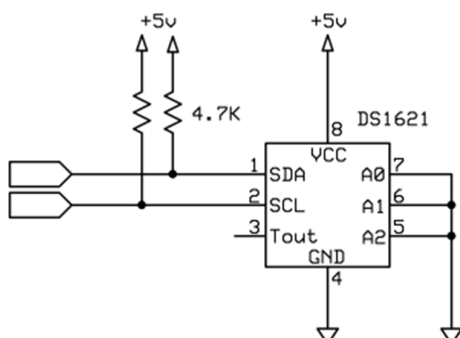


Address คือ 1101000

9

การทดลอง Temperature Sensor DS1621

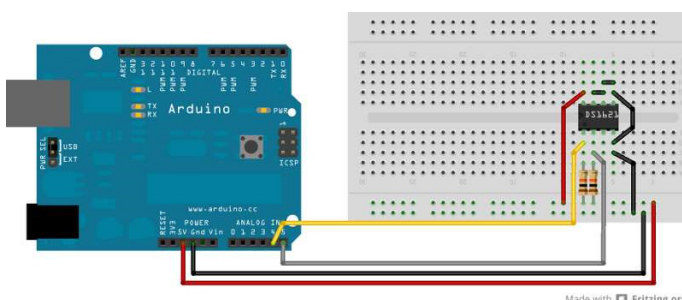
- ตัวอย่างตามรูป



รายละเอียดขา

Address คือ 1001A₂A₁A₀

หรือ 48H-4FH



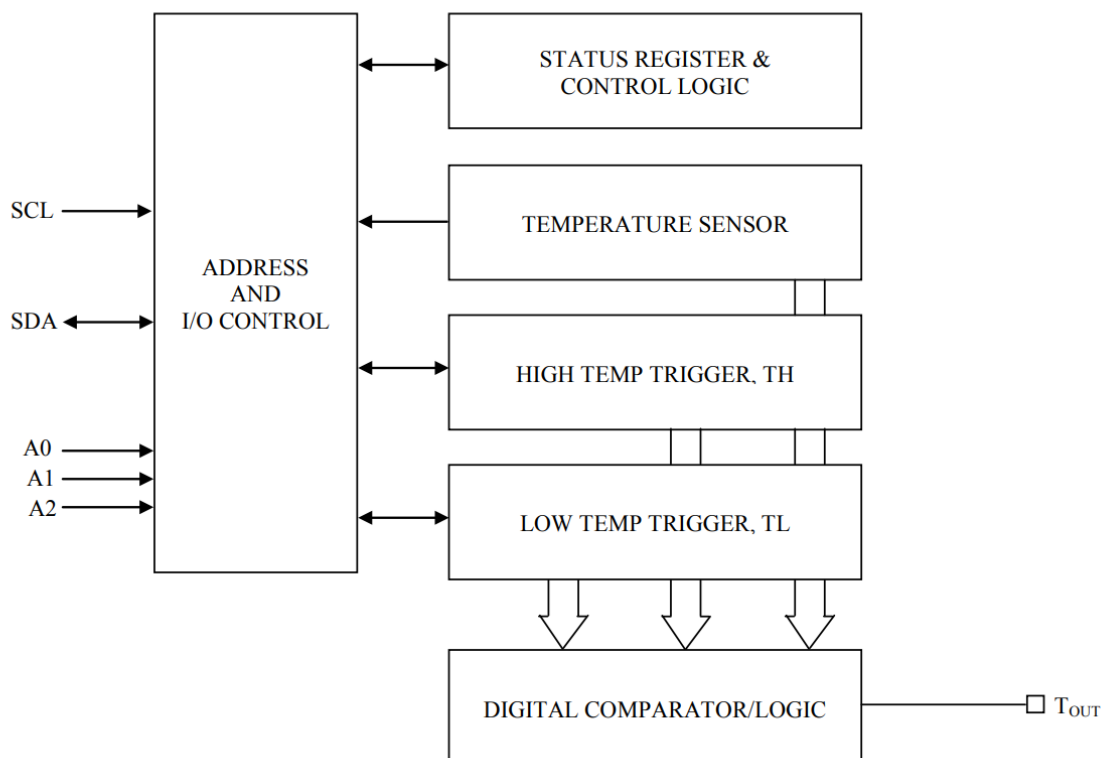
รายละเอียด IC เบอร์ DS1621

ผู้ผลิตคือ Dallas Semiconductor เป็นไอซีวัดอุณหภูมิที่ใช้การติดต่อผ่านระบบบัส I²C สำหรับคุณสมบัติที่สำคัญของ DS1621 มีดังนี้

- สามารถวัดอุณหภูมิ ได้ตั้งแต่ -55 เซลเซียส ถึง +125 เซลเซียส โดยมีความละเอียดในการวัด 0.5 เซลเซียส
- ความละเอียดของข้อมูลอุณหภูมิดิจิทัล 9 บิต
- ใช้ไฟเลี้ยงได้ตั้งแต่ 2.7-5.5 โวลท์
- ใช้เวลาในการวัดอุณหภูมิแล้วแปลงเป็นข้อมูลดิจิทัล 1 วินาที
- สามารถทำงานเป็นเทอร์โมสแตต (thermostat) ได้ พร้อมขาเอาต์พุต 1 ขา
- สามารถตั้งค่าอุณหภูมิที่ต้องการเมื่อทำงานเป็นเทอร์โมสแตตได้ โดยติดต่อผ่านระบบบัส I²C และค่าที่กำหนดนี้จะคงอยู่ตลอดไป แม้ปลดไฟเลี้ยงแล้วก็ตามสามารถเปลี่ยนแปลงได้โดยการ กำหนดทางซอฟต์แวร์เท่านั้น
- สามารถต่อพ่วงกันได้สูงสุด 8 ตัว

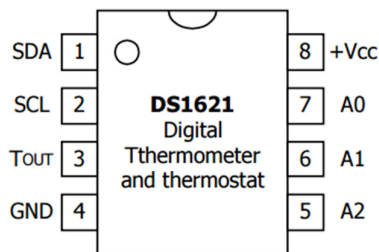
11

Figure 1. DS1621 FUNCTIONAL BLOCK DIAGRAM



12

รายละเอียดขา และการแปลงข้อมูล



ขาที่	ชื่อขา	หน้าที่/การทำงาน
1	SDA	ขาข้อมูลอนุกรมสำหรับเชื่อมต่อกับระบบบัส I ² C
2	SCL	ขาสัญญาณนาฬิกาสำหรับเชื่อมต่อกับระบบบัส I ² C
3	TOUT	ขาเอาต์พุตเทอร์โมสแตต กระแสซอร์สเอาต์พุต 1mA กระแสซิงก์เอาต์พุต 4mA "เอกทีฟ" เมื่ออุณหภูมิสูงถึงจุดกระตุ้นสูง (TH) "รีเซต" เมื่ออุณหภูมิลดลงต่ำกว่าจุดกระตุ้นต่ำ (TL)
4	GND	ขาต่อกราวด์ของวงจร
5-7	A2-A0	ขากำหนดแอดเดรสของ DS1621 หากมีตัวเดียวในระบบควรต่อลงกราวด์
8	+Vcc	ขาต่อไฟเลี้ยง +5V

อุณหภูมิ	ข้อมูลดิจิทัลเอาต์พุต (เลขฐานสอง)	ข้อมูลดิจิทัลเอาต์พุต (เลขฐานสิบหก)
+125°C	01111101 00000000	7B00H
+25°C	00011001 00000000	1900H
+0.5°C	00000001 00000000	0080H
0°C	00000000 00000000	0000H
-0.5°C	11111111 10000000	FF80H
-25°C	11100111 00000000	E700H
-55°C	11001001 00000000	C900H

13

รีจิสเตอร์ภายในไอซี DS1621

OPERATION AND CONTROL

The DS1621 must have temperature settings resident in the TH and TL registers for thermostatic operation. A configuration/status register also determines the method of operation that the DS1621 will use in a particular application, as well as indicating the status of the temperature conversion operation.

The configuration register is defined as follows:

MSb	Bit 6	Bit5	Bit 4	Bit 3	Bit 2	Bit 1	LSb
DONE	THF	TLF	NVB	X	X	POL	1SHOT

where

DONE = Conversion Done bit. "1" = Conversion complete, "0" = Conversion in progress.

THF = Temperature High Flag. This bit will be set to "1" when the temperature is greater than or equal to the value of TH. It will remain "1" until reset by writing "0" into this location or removing power from the device. This feature provides a method of determining if the DS1621 has ever been subjected to temperatures above TH while power has been applied.

TLF = Temperature Low Flag. This bit will be set to “1” when the temperature is less than or equal to the value of TL. It will remain “1” until reset by writing “0” into this location or removing power from the device. This feature provides a method of determining if the DS1621 has ever been subjected to temperatures below TL while power has been applied.

NVB = Nonvolatile Memory Busy flag. “1” = Write to an E² memory cell in progress, “0” = nonvolatile memory is not busy. A copy to E² may take up to 10 ms.

POL = Output Polarity Bit. “1” = active high, “0” = active low. This bit is nonvolatile.

1SHOT = One Shot Mode. If 1SHOT is “1”, the DS1621 will perform one temperature conversion upon receipt of the Start Convert T protocol. If 1SHOT is “0”, the DS1621 will continuously perform temperature conversions. This bit is nonvolatile.

X = Reserved.

For typical thermostat operation the DS1621 will operate in continuous mode. However, for applications where only one reading is needed at certain times or to conserve power, the one-shot mode may be used. Note that the thermostat output (T_{OUT}) will remain in the state it was in after the last valid temperature conversion cycle when operating in one-shot mode.

คำสั่งการทำงาน

คำสั่ง	ข้อมูลคำสั่ง	การทำงานหลังส่งคำสั่ง	หมายเหตุ
คำสั่งเกี่ยวกับการแปลงค่าอุณหภูมิ			
อ่านค่าอุณหภูมิ	AAH	อ่านข้อมูล 2 ไบต์	
อ่านค่าตัวนับ	A8H	อ่านข้อมูล 1 ไบต์	
อ่านค่าตัวนับสไลป	A9H	อ่านข้อมูล 1 ไบต์	
เริ่มต้นแปลงค่าอุณหภูมิ	EEH	หยุดและเตรียมพร้อมเริ่มต้น	1
หยุดแปลงค่าอุณหภูมิ	22H	หยุดและเตรียมพร้อมเริ่มต้น	1
คำสั่งในโหมดเทอร์โมสตัต			
เข้าถึง TH	A1H	เขียนข้อมูล 1 ไบต์	2
เข้าถึง TL	A2H	เขียนข้อมูล 1 ไบต์	2
เข้าถึงรีจิสเตอร์ Config.	ACH	เขียนข้อมูล 1 ไบต์	2

โปรแกรมอ่านค่าจาก DS1621

```
#include <Wire.h>

void setup()
{
    Serial.begin(9600);

    Wire.begin();
    Wire.beginTransmission(0x48);           // connect to DS1621 (#0)
    Wire.write(0xAC);                       // Access Config
    Wire.write(0x02);                       // set for continuous conversion
    Wire.beginTransmission(0x48);          // restart
    Wire.write(0xEE);                       // start conversions
    Wire.endTransmission();
}
```

17

```
void loop()
{
    int8_t firstByte;
    int8_t secondByte;
    float temp = 0;

    delay(1000);                           //give time for measurement

    Wire.beginTransmission(0x48);
    Wire.write(0xAA);                       //read temperature command
    Wire.endTransmission();
    Wire.requestFrom(0x48, 2);              //request two bytes from DS1621 (0.5 deg. resolution)

    firstByte = Wire.read();                //get first byte
    secondByte = Wire.read();               //get second byte

    temp = firstByte;

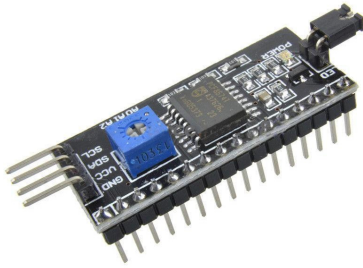
    if (secondByte)                         //if there is a 0.5 deg difference
        temp += 0.5;

    Serial.println(temp);
}
```

18

ศึกษาด้วยตนเอง เรื่อง I2C LCD

- เนื่องจากการต่อจอ LCD ต้องใช้พอร์ตนับจำนวนมาก ถึง 6 พอร์ต ทำให้เมื่อนำไปต่ออุปกรณ์อื่นๆ พอร์ตอาจจะไม่เพียงพอ จึงมีผู้ผลิตตัวแปลง เพื่อให้ Arduino สามารถติดต่อกับจอ LCD โดยผ่าน I2C ได้ โดยใช้พอร์ตเพียง 2 พอร์ตคือขา A4 และ A5
- ให้ สุดยอดนักศึกษาของคณะ ศึกษาและเขียนโปรแกรม ให้ **แสดงผลอุณหภูมิที่วัดได้ ออกจอ LCD ที่ต่อผ่าน I2C**
- Hint หา address ของอุปกรณ์ ให้เจอก่อน



<https://opencircuit.nl/ProductInfo/1000061/I2C-LCD-interface.pdf>