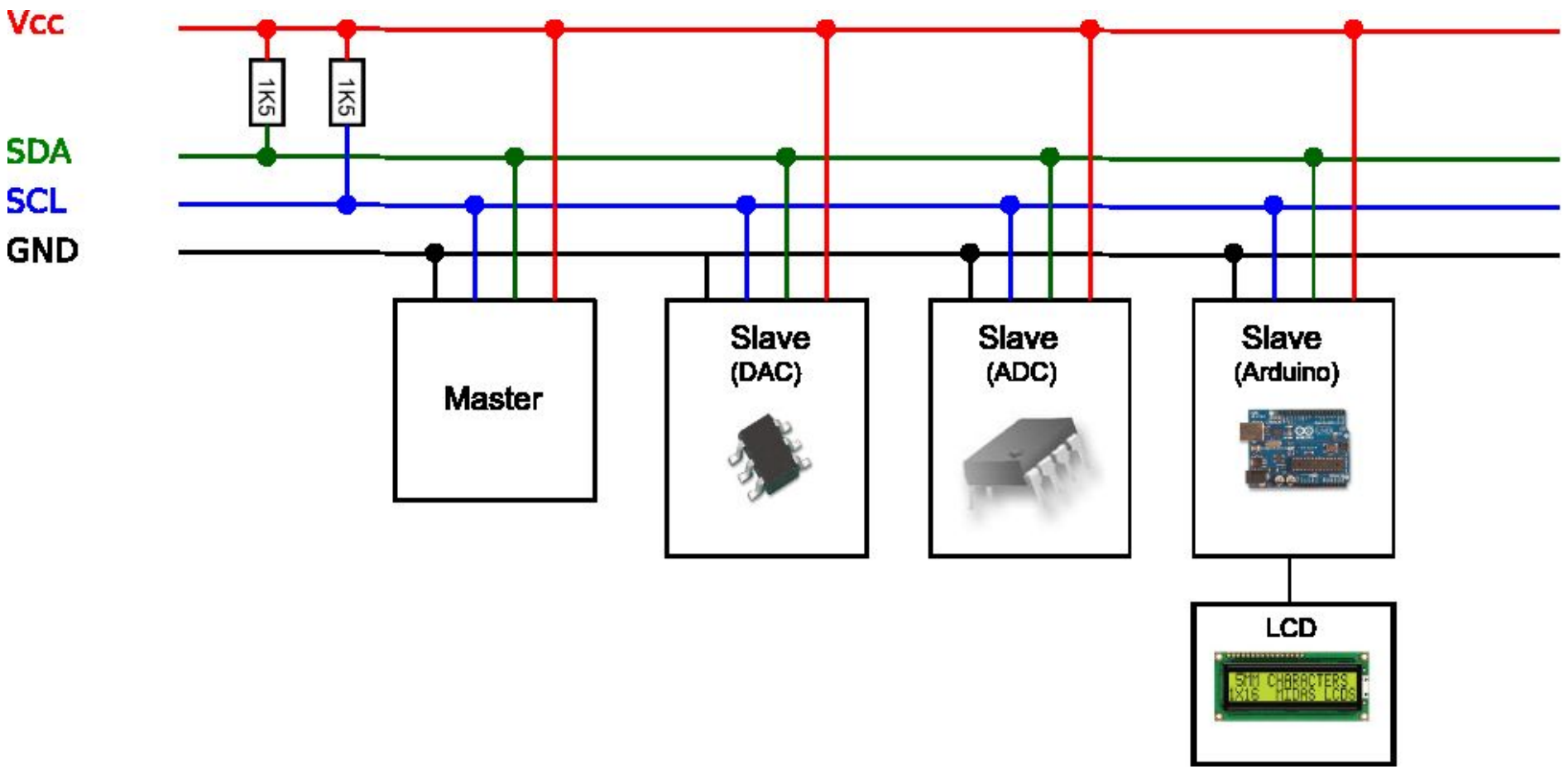
Лекция 12

Интерфейсы передачи данных

Интерфейсная шина I2C



Взаимодействие и идентификация устройств

PART NO.	XX	-XX	X	XX
Device	Address Options	Supply Voltage	Operating Temperature	Package

Device:	TC74: Serial Digital Thermal Sensor
Address Options:	A0 = 1001 000 A1 = 1001 001 A2 = 1001 010 A3 = 1001 011 A4 = 1001 100 A5 = 1001 101 * A6 = 1001 110 A7 = 1001 111 * Default Address
Output Voltage:	3.3 = Accuracy optimized for 3.3V 5.0 = Accuracy optimized for 5.0V
Operating Temperature:	V = -40°C ≤ T_A ≤ +125°C
Package:	AT = TO-220-5

Examples:

a) TC74A0-3.3VAT: TO-220 Serial Digital Thermal Sensor
b) TC74A1-3.3VAT: TO-220 Serial Digital Thermal Sensor
c) TC74A2-3.3VAT: TO-220 Serial Digital Thermal Sensor
d) TC74A3-3.3VAT: TO-220 Serial Digital Thermal Sensor
e) TC74A4-3.3VAT: TO-220 Serial Digital Thermal Sensor
f) TC74A5-3.3VAT: TO-220 Serial Digital Thermal Sensor *
g) TC74A6-3.3VAT: TO-220 Serial Digital Thermal Sensor
h) TC74A7-3.3VAT: TO-220 Serial Digital Thermal Sensor

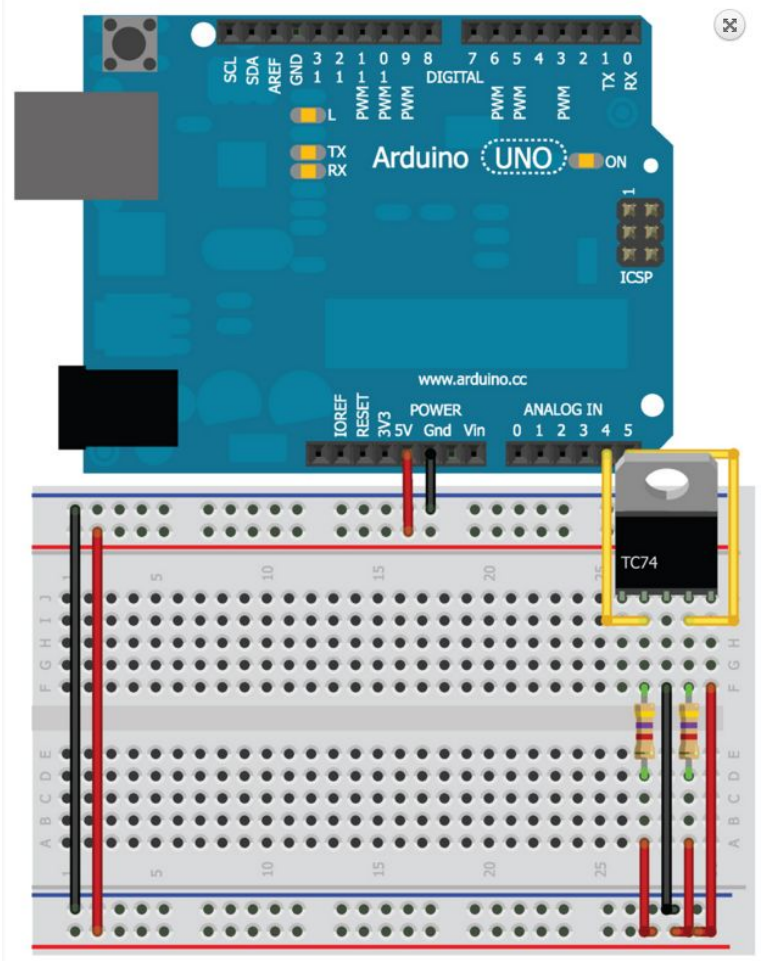
a) TC74A0-5.0VAT: TO-220 Serial Digital Thermal Sensor
b) TC74A1-5.0VAT: TO-220 Serial Digital Thermal Sensor
c) TC74A2-5.0VAT: TO-220 Serial Digital Thermal Sensor
d) TC74A3-5.0VAT: TO-220 Serial Digital Thermal Sensor
e) TC74A4-5.0VAT: TO-220 Serial Digital Thermal Sensor
f) TC74A5-5.0VAT: TO-220 Serial Digital Thermal Sensor *
g) TC74A6-5.0VAT: TO-220 Serial Digital Thermal Sensor
h) TC74A7-5.0VAT: TO-220 Serial Digital Thermal Sensor

* Default Address

Основные этапы управления I2C-устройством

- Мастер посылает стартовый бит.
- Мастер посылает 7-разрядный адрес ведомого устройства.
- Мастер устанавливает на шине данных «1»(чтение) или «0» (запись).
- Ведомое устройство выставляет бит АСК (низкий логический уровень).
- В режиме записи, мастер передает один байт информации, ведомое устройство выдает бит АСК. В режиме чтения, мастер получает один байт информации и посылает бит АСК в ведомое после каждого байта.
- Когда связь завершена, мастер посылает стоп-бит.

Соединение платы Arduino и цифрового датчика TC74



Протокол обмена датчика TC74

Write Byte Format

S	Address	WR	ACK	Command	ACK	Data	ACK	P
	7 Bits			8 Bits		8 Bits		

Slave Address

Command Byte: selects which register you are writing to.

Data Byte: data goes into the register set by the command byte.

Read Byte Format

S	Address	WR	ACK	Command	ACK	S	Address	RD	ACK	Data	NACK	P
	7 Bits			8 Bits			7 Bits			8 Bits		

Slave Address

Command Byte: selects which register you are reading from.

Slave Address: repeated due to change in data-flow direction.

Data Byte: reads from the register set by the command byte.

Receive Byte Format

S	Address	RD	ACK	Data	NACK	P
	7 Bits			8 Bits		

S = START Condition
P = STOP Condition
Shaded = Slave Transmission

Data Byte: reads data from the register commanded by the last Read Byte or Write Byte transmission.

4.0 REGISTER SET AND PROGRAMMER'S MODEL

TABLE 4-1: COMMAND BYTE DESCRIPTION (SMBUS/I²C READ_BYTE AND WRITE_BYTE)

Command	Code	Function
RTR	00h	Read Temperature (TEMP)
RWCR	01h	Read/Write Configuration (CONFIG)

TABLE 4-2: CONFIGURATION REGISTER (CONFIG); 8 BITS, READ/WRITE)

Bit	POR	Function	Type	Operation
D[7]	0	STANDBY Switch	Read/ Write	1 = standby, 0 = normal
D[6]	0	Data Ready *	Read Only	1 = ready 0 = not ready
D[5]-D[0]	0	Reserved - Always returns zero when read	N/A	N/A

Note 1: *DATA_RDY bit RESET at power-up and SHDN enable.

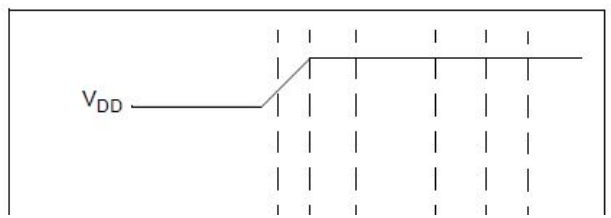


TABLE 4-3: TEMPERATURE REGISTER (TEMP)

D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
MSB	X	X	X	X	X	X	LSB

In temperature data registers, each unit value represents one degree (Celsius). The value is in 2's complement binary format such that a reading of 0000 0000b corresponds to 0°C. Examples of this temperature to binary value relationship are shown in Table 4-4.

TABLE 4-4: TEMPERATURE-TO-DIGITAL VALUE CONVERSION (TEMP)

Actual Temperature	Registered Temperature	Binary Hex
+130.00°C	+127°C	0111 1111
+127.00°C	+127°C	0111 1111
+126.50°C	+126°C	0111 1110
+25.25°C	+25°C	0001 1001
+0.50°C	0°C	0000 0000
+0.25°C	0°C	0000 0000
0.00°C	0°C	0000 0000
-0.25°C	-1°C	1111 1111
-0.50°C	-1°C	1111 1111
-0.75°C	-1°C	1111 1111
-1.00°C	-1°C	1111 1111
-25.00°C	-25°C	1110 0111
-25.25°C	-26°C	1110 0110
-54.75°C	-55°C	1100 1001
-55.00°C	-55°C	1100 1001
-65.00°C	-65°C	1011 1111

Программа (часть 1)

```
//Include Wire I2C library
#include <Wire.h>
int temp_address = 72; //1001000 written as decimal number

void setup()
{
  //Start serial communication at 9600 baud
  Serial.begin(9600);

  //Create a Wire object
  Wire.begin();
}
```

Программа (часть 2)

```
void loop()
{
  //Send a request
  //Start talking to the device at the specified address
  Wire.beginTransmission(temp_address);
  //Send a bit asking for register zero, the data register
  Wire.write(0);
  //Complete Transmission
  Wire.endTransmission();
```

Программа (часть 3)

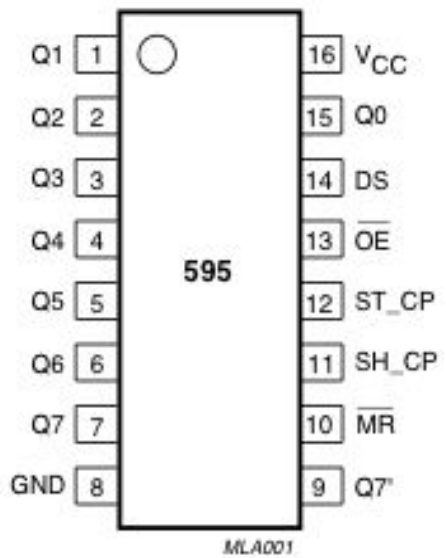
```
//Read the temperature from the device
  //Request 1 Byte from the specified address
  Wire.requestFrom(temp_address, 1);
  //wait for response
  while(Wire.available() == 0);
  // Get the temp and read it into a variable
  int c = Wire.read();

  //Do some math to convert the Celsius to Fahrenheit
  int f = round(c*9.0/5.0 +32.0);
```

Программа (часть 4)

```
 //Send the temperature in degrees C and F to the serial monitor
  Serial.print(c);
  Serial.print("C ");
  Serial.print(f);
  Serial.println("F");


  delay(500);
}
```

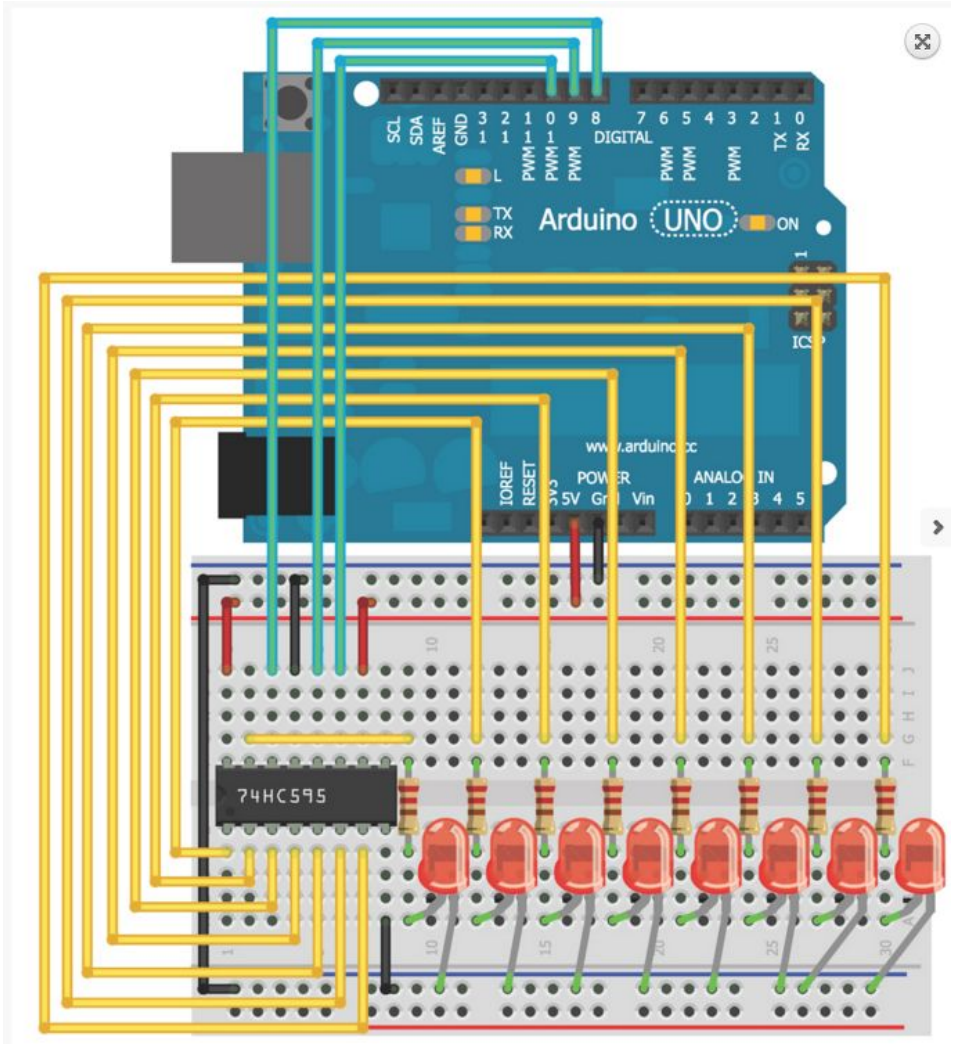
Сдвиговый регистр 74HC595

Распиновка входов/выходов регистра



Пины 1-7, 15	Q0 " Q7	Параллельные выходы
Пин 8	GND	Земля
Пин 9	Q7"	Выход для последовательного соединения регистров
Пин 10	MR	Сброс значений регистра. Сброс происходит при получение LOW
Пин 11	SH_CP	Вход для тактовых импульсов
Пин 12	ST_CP	Синронизация ("защелкивание") выходов
Пин 13	OE	Вход для переключения состояния выходов из высокоомного в рабочее
Пин 14	DS	Вход для последовательных данных
Пин 16	Vcc	Питание

Подключение 8 светодиодов



Программа (часть)

```
const int SER   =8;    //Serial Output to Shift Register
const int LATCH =9;    //Shift Register Latch Pin
const int CLK   =10;   //Shift Register Clock Pin

void setup()
{
  //Set pins as outputs
  pinMode(SER, OUTPUT);
  pinMode(LATCH, OUTPUT);
  pinMode(CLK, OUTPUT);
```

```
digitalWrite(LATCH, LOW);               //Latch Low
  shiftOut(SER, CLK, MSBFIRST, B10101010); //Shift Most Sig. Bit First
  digitalWrite(LATCH, HIGH);             //Latch High - Show pattern
}

void loop()
{
  //Do nothing
}
```

Преобразование между двоичными и десятичными форматами

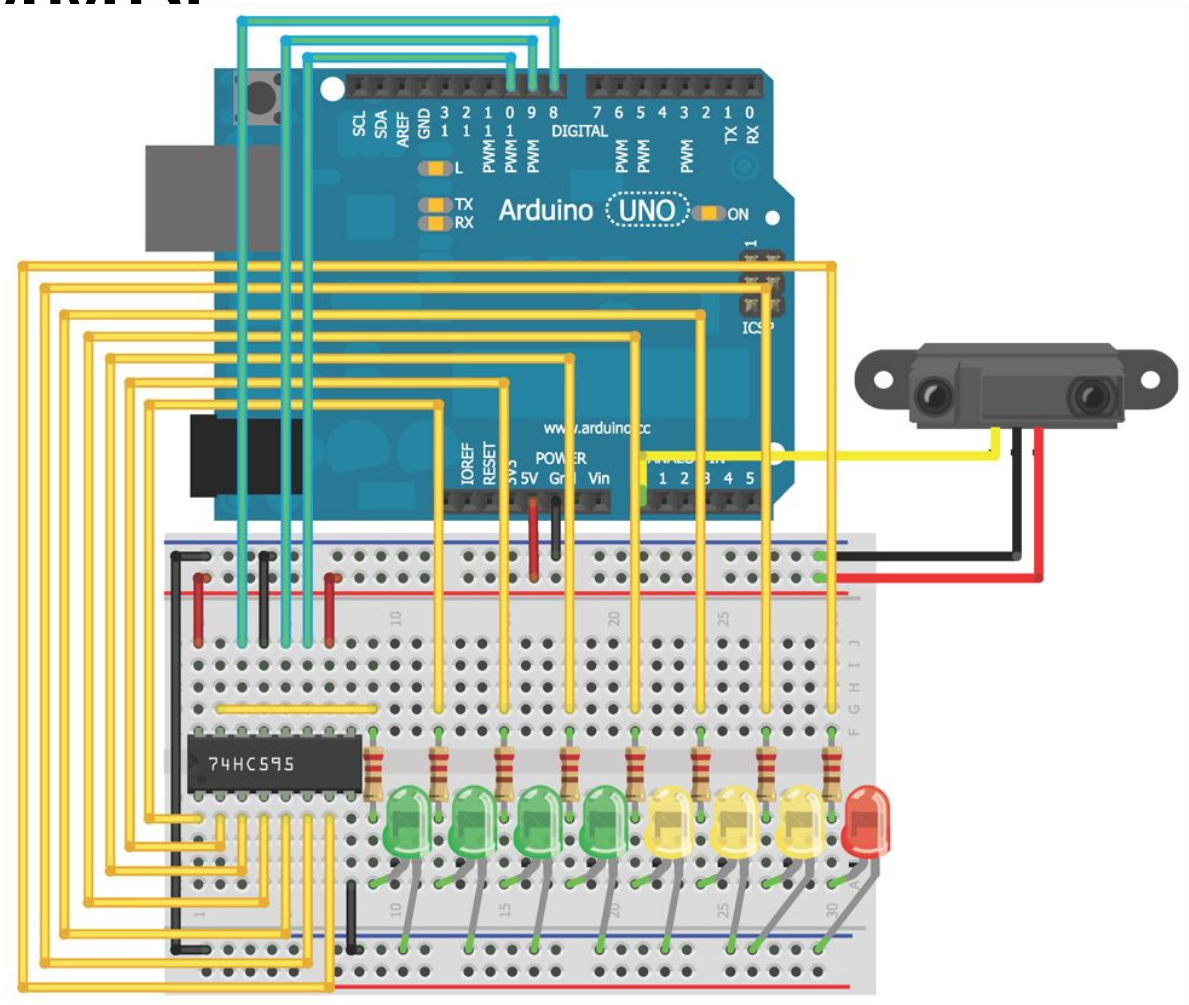
Заменим:

shiftOut(SER, CLK, MSBFIRST, B10101010);

На:

shiftOut(DATA, CLOCK, MSBFIRST, 170);

Отображение данных в виде гистограммы



Программа

```
const int SER   =8;    //Serial Output to Shift Register
const int LATCH =9;    //Shift Register Latch Pin
const int CLK   =10;   //Shift Register Clock Pin
const int DIST  =0;    //Distance Sensor on Analog Pin 0

//Possible LED settings
int vals[9] = {0,1,3,7,15,31,63,127,255};

//Maximum value provided by sensor
int maxVal = 500;

//Minimum value provided by sensor
int minVal = 0;
```

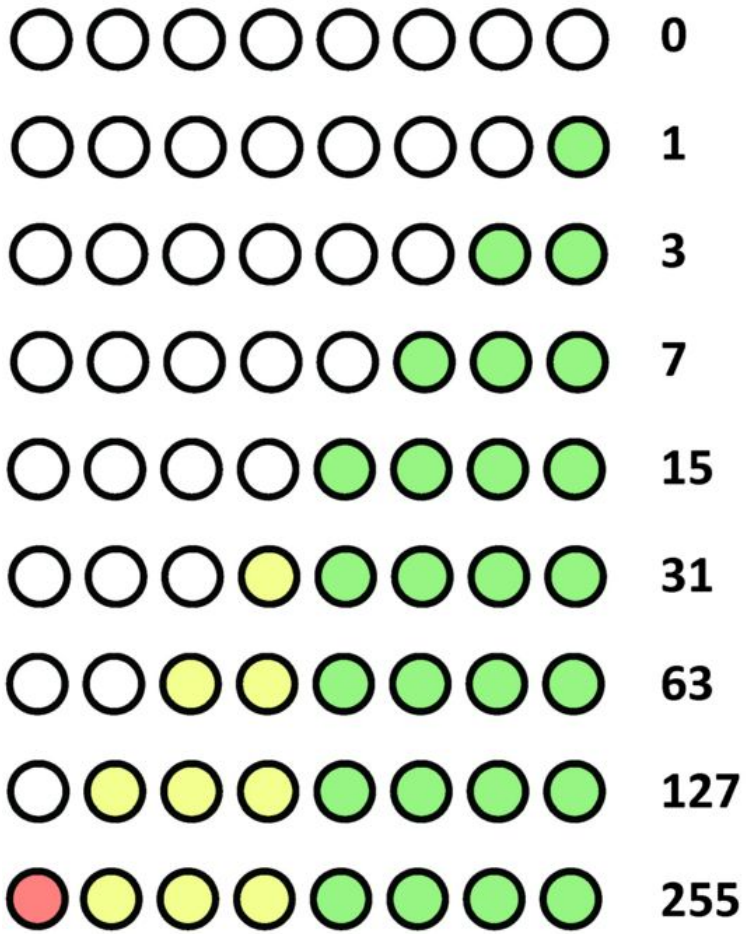
Программа (часть 2)

```
void setup()
{
  //Set pins as outputs
  pinMode(SER, OUTPUT);
  pinMode(LATCH, OUTPUT);
  pinMode(CLK, OUTPUT);
}
```

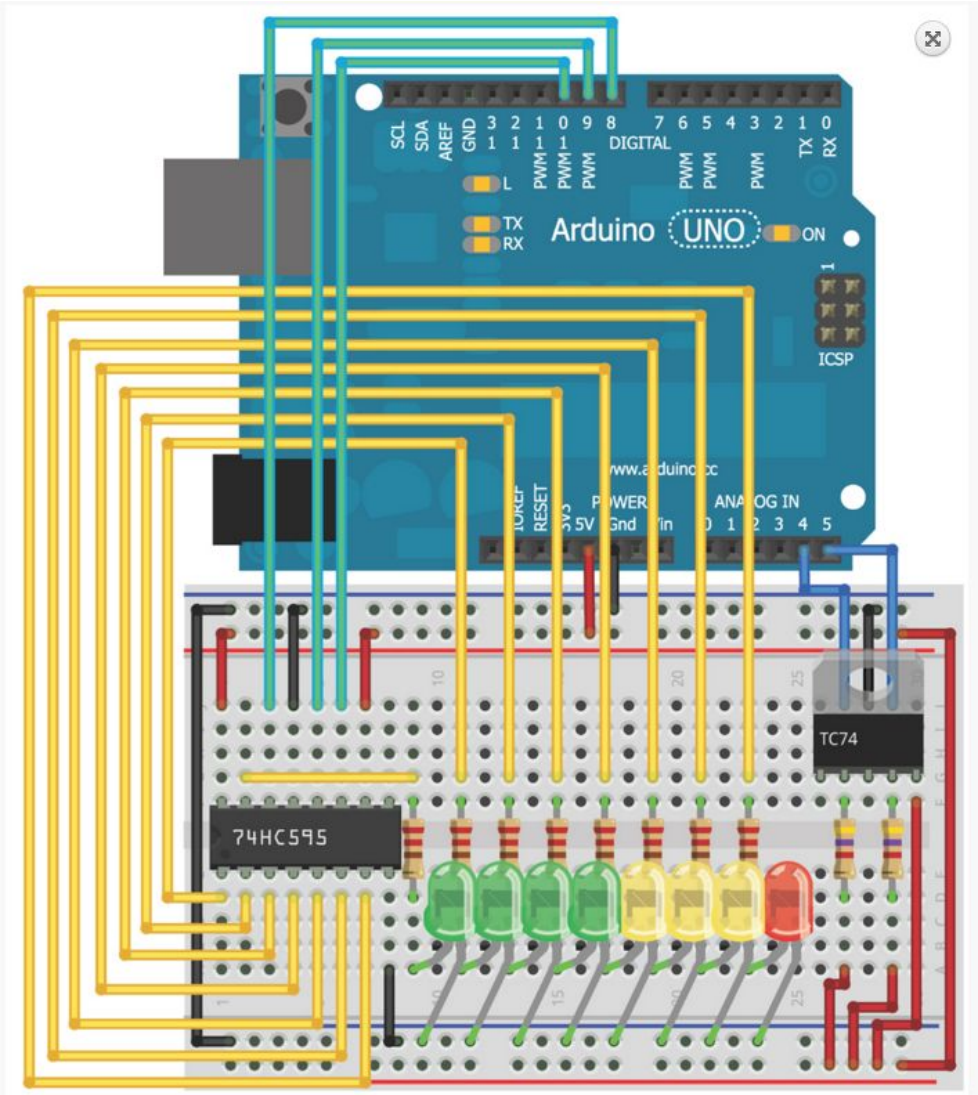
Программа (часть 3)

```
void loop()
{
  int distance = analogRead(DIST);
  distance = map(distance, minVal, maxVal, 0, 8);
  distance = constrain(distance,0,8);

  digitalWrite(LATCH, LOW);               //Latch low - start sending
  shiftOut(SER, CLK, MSBFIRST, vals[distance]); //Send data, MSB first
  digitalWrite(LATCH, HIGH);              //Latch high - stop sending
  delay(10);                          //Animation speed

}
```

Комбинации светодиодов и соответствующие им значения



Регистр сдвига и I2C



Программа (часть 1)

```
//Include Wire I2C library
#include <Wire.h>

const int SER   =8;  //Serial Output to Shift Register
const int LATCH =9;  //Shift Register Latch Pin
const int CLK   =10; //Shift Register Clock Pin

int temp_address = 72;

//Possible LED settings
int vals[8] = {1,3,7,15,31,63,127,255};
```

Программа (часть 2)

```
void setup()
{
  //Instantiate serial communicatuion at 9600 bps
  Serial.begin(9600);

  //Create a Wire Object
  Wire.begin();

  //Set shift register pins as outputs
  pinMode(SER, OUTPUT);
  pinMode(LATCH, OUTPUT);
  pinMode(CLK, OUTPUT);
}
```

Программа (часть 3)

```
void loop()
{
  //Send a request
  //Start talking to the device at the specified address
  Wire.beginTransmission(temp_address);
  //Send a bit asking for register zero, the data register
  Wire.write(0);
  //Complete Transmission
  Wire.endTransmission();
```

Программа (часть 4)

```
//Read the temperature from the device
  //Request 1 Byte from the specified address
  Wire.requestFrom(temp_address, 1);
  //wait for response
  while(Wire.available() == 0);
  // Get the temp and read it into a variable
  int c = Wire.read();
```

Программа (часть 5)

```
//Map the temperatures to LED settings
  int graph = map(c, 24, 31, 0, 7);
  graph = constrain(graph,0,7);


  digitalWrite(LATCH, LOW);       //Latch low - start sending data
  shiftOut(SER, CLK, MSBFIRST, vals[graph]); //Send data, most significant bit first
  digitalWrite(LATCH, HIGH);      //Latch high - stop sending data


  //Do some math to convert the Celsius to Fahrenheit
  int f = round(c*9.0/5.0 +32.0);
```

Программа (часть 6)

```
Serial.print(c);
  Serial.print("C,");
  Serial.print(f);
  Serial.print("F.");

  delay(500);
}
```

Отображение данных на мониторе (Processing)

```
import processing.serial.*;
Serial port;
String temp_c = "";
String temp_f = "";
String data = "";
int index = 0;
PFont font;
```

Программа (часть 2)

```
void setup()
{
  size(400,400);
//Change "COM9" to the name of the serial port on your computer
  port = new Serial(this, "COM9", 9600);
  port.bufferUntil('.');
//Change the font name to reflect the name of the font you created
  font = loadFont("AgencyFB-Bold-200.vlw");
  textFont(font, 200);
}
```

Программа (часть 3)

```
void draw()
{
  background(0,0,0);
  fill(46, 209, 2);
  text(temp_c, 70, 175);
  fill(0, 102, 153);
  text(temp_f, 70, 370);
}
```

Программа (часть 4)

```
void serialEvent (Serial port)
{
  data = port.readStringUntil('.');
  data = data.substring(0, data.length() - 1);

  //Look for the comma between Celcius and Farenheit
  index = data.indexOf(",");
  // fetch the C Temp
  temp_c = data.substring(0, index);
  //Fetch the F temp
  temp_f = data.substring(index+1, data.length());
}
```