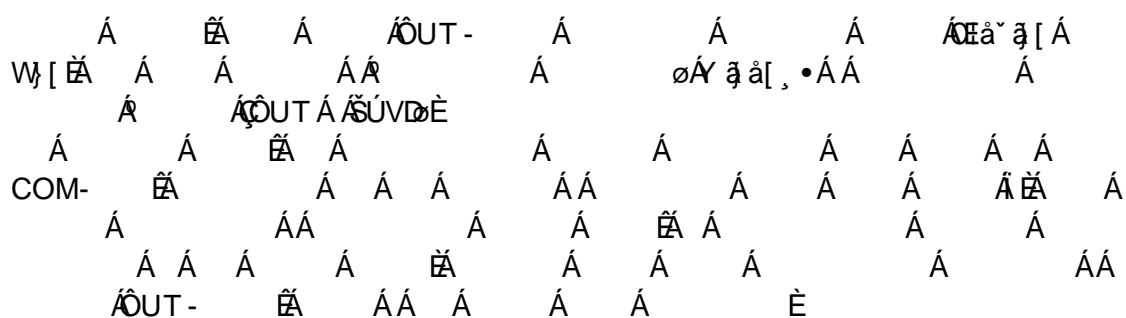
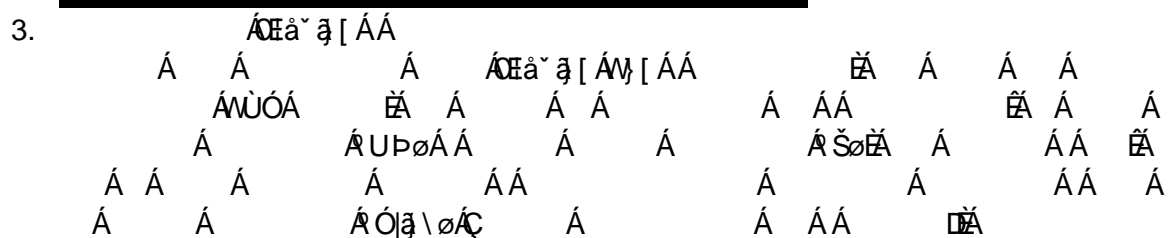


%

Robstore.ru



1. *Journal of the American Medical Association*, 2000; 284: 2689-2695.

1. *Journal of Management Studies*, 1996, 33(1), 1-14.
 2. *Journal of Management Studies*, 1996, 33(1), 15-30.
 3. *Journal of Management Studies*, 1996, 33(1), 31-46.
 4. *Journal of Management Studies*, 1996, 33(1), 47-62.
 5. *Journal of Management Studies*, 1996, 33(1), 63-78.
 6. *Journal of Management Studies*, 1996, 33(1), 79-94.
 7. *Journal of Management Studies*, 1996, 33(1), 95-110.
 8. *Journal of Management Studies*, 1996, 33(1), 111-126.
 9. *Journal of Management Studies*, 1996, 33(1), 127-142.
 10. *Journal of Management Studies*, 1996, 33(1), 143-158.
 11. *Journal of Management Studies*, 1996, 33(1), 159-174.
 12. *Journal of Management Studies*, 1996, 33(1), 175-190.
 13. *Journal of Management Studies*, 1996, 33(1), 191-206.
 14. *Journal of Management Studies*, 1996, 33(1), 207-222.
 15. *Journal of Management Studies*, 1996, 33(1), 223-238.
 16. *Journal of Management Studies*, 1996, 33(1), 239-254.
 17. *Journal of Management Studies*, 1996, 33(1), 255-270.
 18. *Journal of Management Studies*, 1996, 33(1), 271-286.
 19. *Journal of Management Studies*, 1996, 33(1), 287-302.
 20. *Journal of Management Studies*, 1996, 33(1), 303-318.
 21. *Journal of Management Studies*, 1996, 33(1), 319-334.
 22. *Journal of Management Studies*, 1996, 33(1), 335-350.
 23. *Journal of Management Studies*, 1996, 33(1), 351-366.
 24. *Journal of Management Studies*, 1996, 33(1), 367-382.
 25. *Journal of Management Studies*, 1996, 33(1), 383-398.
 26. *Journal of Management Studies*, 1996, 33(1), 399-414.
 27. *Journal of Management Studies*, 1996, 33(1), 415-430.
 28. *Journal of Management Studies*, 1996, 33(1), 431-446.
 29. *Journal of Management Studies*, 1996, 33(1), 447-462.
 30. *Journal of Management Studies*, 1996, 33(1), 463-478.
 31. *Journal of Management Studies*, 1996, 33(1), 479-494.
 32. *Journal of Management Studies*, 1996, 33(1), 495-510.
 33. *Journal of Management Studies*, 1996, 33(1), 511-526.
 34. *Journal of Management Studies*, 1996, 33(1), 527-542.
 35. *Journal of Management Studies*, 1996, 33(1), 543-558.
 36. *Journal of Management Studies*, 1996, 33(1), 559-574.
 37. *Journal of Management Studies*, 1996, 33(1), 575-590.
 38. *Journal of Management Studies*, 1996, 33(1), 591-606.
 39. *Journal of Management Studies*, 1996, 33(1), 607-622.
 40. *Journal of Management Studies*, 1996, 33(1), 623-638.
 41. *Journal of Management Studies*, 1996, 33(1), 639-654.
 42. *Journal of Management Studies*, 1996, 33(1), 655-670.
 43. *Journal of Management Studies*, 1996, 33(1), 671-686.
 44. *Journal of Management Studies*, 1996, 33(1), 687-702.
 45. *Journal of Management Studies*, 1996, 33(1), 703-718.
 46. *Journal of Management Studies*, 1996, 33(1), 719-734.
 47. *Journal of Management Studies*, 1996, 33(1), 735-750.
 48. *Journal of Management Studies*, 1996, 33(1), 751-766.
 49. *Journal of Management Studies*, 1996, 33(1), 767-782.
 50. *Journal of Management Studies*, 1996, 33(1), 783-798.
 51. *Journal of Management Studies*, 1996, 33(1), 799-814.
 52. *Journal of Management Studies*, 1996, 33(1), 815-830.
 53. *Journal of Management Studies*, 1996, 33(1), 831-846.
 54. *Journal of Management Studies*, 1996, 33(1), 847-862.
 55. *Journal of Management Studies*, 1996, 33(1), 863-878.
 56. *Journal of Management Studies*, 1996, 33(1), 879-894.
 57. *Journal of Management Studies*, 1996, 33(1), 895-910.
 58. *Journal of Management Studies*, 1996, 33(1), 911-926.
 59. *Journal of Management Studies*, 1996, 33(1), 927-942.
 60. *Journal of Management Studies*, 1996, 33(1), 943-958.
 61. *Journal of Management Studies*, 1996, 33(1), 959-974.
 62. *Journal of Management Studies*, 1996, 33(1), 975-990.
 63. *Journal of Management Studies*, 1996, 33(1), 991-1006.
 64. *Journal of Management Studies*, 1996, 33(1), 1007-1022.
 65. *Journal of Management Studies*, 1996, 33(1), 1023-1038.
 66. *Journal of Management Studies*, 1996, 33(1), 1039-1054.
 67. *Journal of Management Studies*, 1996, 33(1), 1055-1070.
 68. *Journal of Management Studies*, 1996, 33(1), 1071-1086.
 69. *Journal of Management Studies*, 1996, 33(1), 1087-1102.
 70. *Journal of Management Studies*, 1996, 33(1), 1103-1118.
 71. *Journal of Management Studies*, 1996, 33(1), 1119-1134.
 72. *Journal of Management Studies*, 1996, 33(1), 1135-1150.
 73. *Journal of Management Studies*, 1996, 33(1), 1151-1166.
 74. *Journal of Management Studies*, 1996, 33(1), 1167-1182.
 75. *Journal of Management Studies*, 1996, 33(1), 1183-1198.
 76. *Journal of Management Studies*, 1996, 33(1), 1199-1214.
 77. *Journal of Management Studies*, 1996, 33(1), 1215-1230.
 78. *Journal of Management Studies*, 1996, 33(1), 1231-1246.
 79. *Journal of Management Studies*, 1996, 33(1), 1247-1262.
 80. *Journal of Management Studies*, 1996, 33(1), 1263-1278.
 81. *Journal of Management Studies*, 1996, 33(1), 1279-1294.
 82. *Journal of Management Studies*, 1996, 33(1), 1295-1310.
 83. *Journal of Management Studies*, 1996, 33(1), 1311-1326.
 84. *Journal of Management Studies*, 1996, 33(1), 1327-1342.
 85. *Journal of Management Studies*, 1996, 33(1), 1343-1358.
 86. *Journal of Management Studies*, 1996, 33(1), 1359-1374.
 87. *Journal of Management Studies*, 1996, 33(1), 1375-1390.
 88. *Journal of Management Studies*, 1996, 33(1), 1391-1406.
 89. *Journal of Management Studies*, 1996, 33(1), 1407-1422.
 90. *Journal of Management Studies*, 1996, 33(1), 1423-1438.
 91. *Journal of Management Studies*, 1996, 33(1), 1439-1454.
 92. *Journal of Management Studies*, 1996, 33(1), 1455-1470.
 93. *Journal of Management Studies*, 1996, 33(1), 1471-1486.
 94. *Journal of Management Studies*, 1996, 33(1), 1487-1502.
 95. *Journal of Management Studies*, 1996, 33(1), 1503-1518.
 96. *Journal of Management Studies*, 1996, 33(1), 1519-1534.
 97. *Journal of Management Studies*, 1996, 33(1), 1535-1550.
 98. *Journal of Management Studies*, 1996, 33(1), 1551-1566.
 99. *Journal of Management Studies*, 1996, 33(1), 1567-1582.
 100. *Journal of Management Studies*, 1996, 33(1), 1583-1598.
 101. *Journal of Management Studies*, 1996, 33(1), 1599-1614.
 102. *Journal of Management Studies*, 1996, 33(1), 1615-1630.
 103. *Journal of Management Studies*, 1996, 33(1), 1631-1646.
 104. *Journal of Management Studies</*

5.

```
int val ;
int ledpin = 13 ;
void setup ()
    Serial.begin(9600);
    pinMode(ledpin, OUTPUT);
}
void loop ()
```

[illegible]

[REDACTED]

·&-

[illegible]

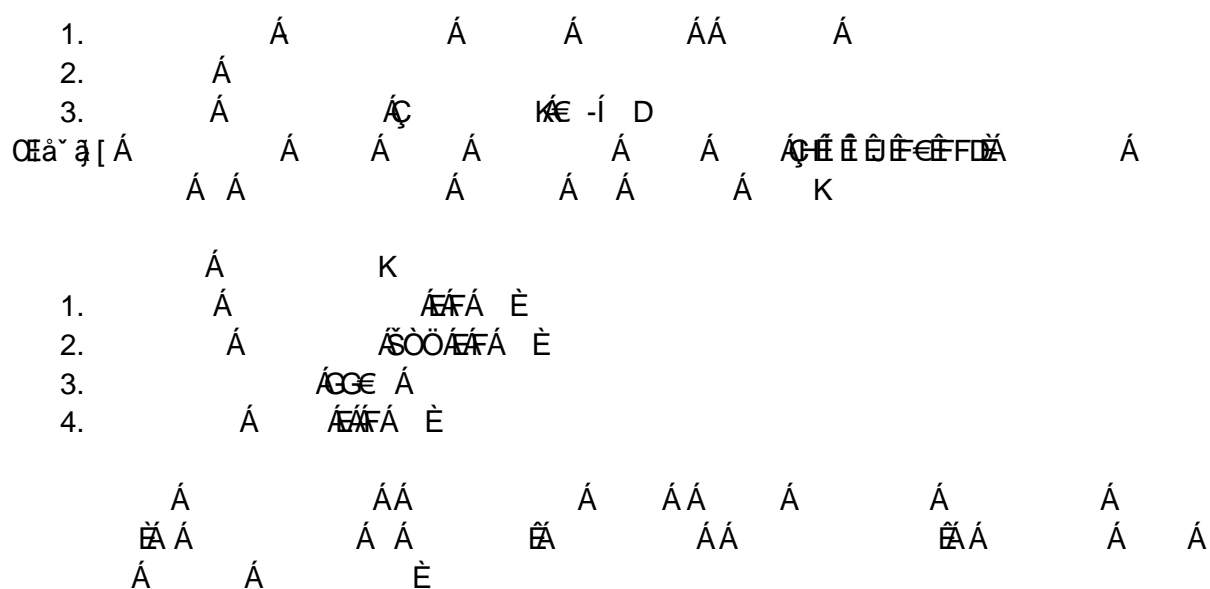
Á Á Á Ę Á Á Á Á Á Ę Á Á

```
void setup ()
{
  ] ă T[ ă^ ă F ă U W ă U W ă D ă D ă Á ă F ă Á ă Á
}
```

```
void loop ()
{
  ăă ăă Y ăă ăă F ăă H ăă P ăă Ő ăă D ăă Á
  ă^ ă^ ă^ ă^ F ă^ ă^ ă^ ă^ D ă^ ă^ Á
  ăă ăă Y ăă ăă F ăă H ăă S ăă U ăă Y ăă D ăă Á
  ă^ ă^ ă^ ă^ F ă^ ă^ ă^ ă^ D ă^ ă^ Á
}
```

Á Á Á Ę Á Á Á Á Á È

RobStore.ru




```
void loop () {
    digitalWrite (ledPin, LOW);
}
```

[illegible]

1. Š^ÁÁ^K È
2. G^ÁÁ^K È
3. Á Á Á Á

Á Á Á Á Á Á Á Á Á Á

```

int NUM;
void setup ()
{
  for (int i = BASE; i < BASE+NUM; i)
  {
    T[ i ] = i;
  }
}

void loop ()
{
  for (int i = BASE; i < BASE+NUM; i)
  {
    delay (200);
  }
  for (int i = BASE; i < BASE+NUM; i)
  {
    delay (200);
  }
}

```



```

void setup () {
  pinMode (redled, OUTPUT); // OUTPUT
  pinMode (yellowled, OUTPUT); // OUTPUT
  pinMode (greenled, OUTPUT); // OUTPUT
}

void loop () {
  digitalWrite (redled, HIGH) ; //
  delay (1000) ; // 1 .
  digitalWrite (redled, LOW); //
  digitalWrite (yellowled, HIGH) ; //
  delay (200) ; // 0.2 .
  digitalWrite (yellowled, LOW) ; //
  digitalWrite (greenled, HIGH) ; //
  delay (1000) ; // 1 .
  digitalWrite (greenled, LOW) ; //
}

```

Á Á Á Á Á Á Á É

```

+--
    ÁÇ ÈÄ ÐÁ      ÁÁ      Á   Á      Á
    ÈÄ      Á      ÈÄ
        Á
    Á   Á   Á       Á   Ä      Á   Á   Á   Á   Á
    Á     Á   Á   Á~::^!ÈÄ      Á   Á   Á   Á   &   Á   Ä
        ÁÁ      Á      ÈÄ   Á      ÁÁ      Á   Ç   Á   Ä
            Á
        1.   Á           K
            ÁFÁ È
        2.   KÁ È
        3.   Á       ÁFÁ È
        4.   Á       Á   Á       Á
            Á   Á   Á   Á   Á   ÁÁ   ÄÁ   ÁÁ   Á~ä[ÈÄ   Á
            È      Á      Á      Á      Á      Á      Á      Á      Á      Á
            Ë
ä ö~::^!ÁÁÄÁ Á   Á~ä[ Á   Á       Á
void setup ()
{
    ] ä T [ á^á~::^!ÉUWÚWÜWÄÄÁ       Á   ÁÁUWÚW
}
void loop ()
{
    unsigned char i, j ; //
    while (1) {
        for (i = 0; i <80; i ++ ) //
        {
            digitalWrite (buzzer, HIGH) ; //
            delay (1) ; //          1ms
            digitalWrite (buzzer, LOW) ; //
            delay (1) ; //          1ms
        }
        for (i = 0; i <100; i ++ ) //
        {
            digitalWrite (buzzer, HIGH) ; //
            delay (2) ; //          2ms
            digitalWrite (buzzer, LOW) ; //
            ä^|æ ÁÇÁÄÁ   ÁÇ •Á
        }
    }
}

```

1. Á K
Á Á Á É
2. ŠÖÖÁ K Á É
3. Á Á Á É
4. Á Á Á

```
void setup ()
{
    pinMode (8, OUTPUT);
}
```

Á Á Á Á Æ Á Á Á Æ Á

$\tilde{E} \quad A \quad A \quad A \quad A \quad A \quad A \quad A \quad A \quad A \quad A$

Arduino.

1.

2.

3.

4.

5.

6.

Á

```
int ledpin = 1FÁÁÁ
```

int

```

int ledpin = 13;
void setup ()
{
  pinMode(ledpin, OUTPUT);
}
void loop ()
{
  val = digitalRead (inputPin);
  if (val == LOW) //
  {DigitalWrite (ledpin, LOW);}
  else {DigitalWrite (ledpin, HIGH);}
}

```

Robstore.ru

1

```

    pinMode (greenpin, INPUT);
}
void loop ()
{
    red = digitalRead (redpin);
    if (red == LOW) {DigitalWrite (redled, LOW);}
    else {DigitalWrite (redled, HIGH);}
    yellow = digitalRead (yellowpin);
    if (yellow == LOW) {DigitalWrite (yellowled, LOW);}
    else {DigitalWrite (yellowled, HIGH);}
    green = digitalRead (greenpin);
    if (green == LOW) {DigitalWrite (greenled, LOW);}
    else {DigitalWrite (greenled, HIGH);}
}

```

Á Á Á Á Á Á ÁÁ Á Á
 Á Á È

100

- [illegible]

```

void setup ()
{
    pinMode (flame, INPUT); // INPUT
    digitalWrite (Beep, LOW);
}

void loop ()
{
    val = analogRead (flame);
    if (val > 100) {
        digitalWrite (Beep, HIGH);
        delay (1000);
        digitalWrite (Beep, LOW);
    }
}

```

```
    else {digitalWrite (Beep, LOW);}
}
```

Á Á Á Á Á ÁÁ Á È

Robstore.ru


```

{
  Serial.print (vol);
  Serial.println ("V");
  temp = vol;
  delay (1000);
  // 1
}
}

```

•

Á KÁ

```
void setup ()
{
    pinMode (ledPin, OUTPUT);
    Serial.begin (9600);
}

void loop ()
{
    sensorValue = analogRead (sensorPin);
    digitalWrite (ledPin, HIGH);
    delay (sensorValue);
    digitalWrite (ledPin, LOW);
    delay (sensorValue);
    Serial.println (sensorValue, DEC);
}
```

Robstore.ru

16 -

- LM35

■

LM35 -

È

1.

2.

3.

Robstore.ru

·%+·-

·-

·

·

"

Á Á Á Á Ě Á Á ŠT H Ě Á
Á Á Á Á Ě

- Á K
1. Arduino K Ě
2. Á K Ě
3. Ě Ě ŠÖÖÁ
4. GOÁ Á K Ě
5. ŠT H Á Á Á Á
6. Á Á Á Á

K

void setup ()

{

pinMode (13, OUTPUT);

pinMode (12, OUTPUT);

pinMode (11, OUTPUT);

}

void loop ()

{

int vol = analogRead (A0) * (5.0 / 1023.0 * 100); //

Á Á Á Á Á Á

{

LM35

ŠÖÖ

```

        digitalWrite (13, HIGH);
        digitalWrite (12, LOW);
        digitalWrite (11, LOW);
    }
    else if (vol> = 32 && vol <= 40)
    {
        digitalWrite (13, LOW);
        digitalWrite (12, HIGH);
        digitalWrite (11, LOW);
    }
    else if (vol> = 41) //
    {
        digitalWrite (13, LOW);
        digitalWrite (12, LOW);
        digitalWrite (11, HIGH);
    }
}

```

Á Á Æ Á Á Á Á Á Æ Á
 Á Á Á Á È


```

    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
    Á    Á    ĚÁ    Á    ÁÁ    Á    ÁÁ    ĚÁ    ÁÁ    Á    Á
    Á    Á    Á    Á    Á    Ě    Á    Á    Á    Á    Á    Á    Á    Á

    ħ    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
    ħ    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
    ħ    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
    ħ    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
    ħ    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
    ħ    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
    ħ    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
    ħ    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á    Á
void digital_1 (void) //    Á    Á
{
    unsigned char j;
    digitalWrite (c, HIGH) ; //    5 (c)
    digitalWrite (b, HIGH) ; //    b
    for (j = 7; j <= 11; j ++ ) //
        digitalWrite (j, LOW);
    digitalWrite (dp, LOW) ; //    DP
}
void digital_2 (void) //    r 2
{
    unsigned char j;
    digitalWrite (b, HIGH);
    digitalWrite (a, HIGH);
    for (j = 9; j <= 11; j ++ )
        digitalWrite (j, HIGH);
    digitalWrite (dp, LOW);
    digitalWrite (c, LOW);
}

```

```

        digitalWrite (f, LOW);
    }
    void digital_3 (void) // 3
    {
        unsigned char j;
        digitalWrite (g, HIGH);
        digitalWrite (d, HIGH);
        for (j = 5; j <= 7; j + +)
            digitalWrite (j, HIGH);
        digitalWrite (dp, LOW);
        digitalWrite (f, LOW);
        digitalWrite (e, LOW);
    }
    void digital_4 (void) // 4
    {
        digitalWrite (c, HIGH);
        digitalWrite (b, HIGH);
        digitalWrite (f, HIGH);
        digitalWrite (g, HIGH);
        digitalWrite (dp, LOW);
        digitalWrite (a, LOW);
        digitalWrite (e, LOW);
        digitalWrite (d, LOW);
    }
    void digital_5 (void) // 5
    {
        unsigned char j;
        for (j = 7; j <= 9; j + +)
            digitalWrite (j, HIGH);
        digitalWrite (c, HIGH);
        digitalWrite (d, HIGH);
        digitalWrite (dp, LOW);
        digitalWrite (b, LOW);
        digitalWrite (e, LOW);
    }
    void digital_6 (void) // 6
    {
        unsigned char j;
        for (j = 7; j <= 11; j + +)
            digitalWrite (j, HIGH);
        digitalWrite (c, HIGH);
        digitalWrite (dp, LOW);
        digitalWrite (b, LOW);
    }
    void digital_7 (void) // 7
    {
        unsigned char j;
        for (j = 5; j <= 7; j + +)

```


Á K

Á K

~~DA~~ Á

```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;  
int e = 5;  
int f = 6;  
int g = 7;  
int p = 8;  
//  
int d4 = 9;
```

Robstore.ru

```

int d3 = 10;
int d2 = 11;
int d1 = 12;
int
long n = 0;
int x = 100;
void setup ()
void setup ()
{
    pinMode (d1, OUTPUT);
    pinMode (d2, OUTPUT);
    pinMode (d3, OUTPUT);
    pinMode (d4, OUTPUT);
    pinMode (a, OUTPUT);
    pinMode (b, OUTPUT);
    pinMode (c, OUTPUT);
    pinMode (d, OUTPUT);
    pinMode (e, OUTPUT);
    pinMode (f, OUTPUT);
    pinMode (g, OUTPUT);
    pinMode (p, OUTPUT);
}
void loop ()
{
    clearLEDs ();
    pickDigit (1);
    pickNumber ((n/x/1000)% 10);
    delayMicroseconds (del);
    clearLEDs ();
    pickDigit (2);
    pickNumber ((n/x/100)% 10);
    delayMicroseconds (del);
    clearLEDs ();
    pickDigit (3);
    dispDec (3);
    pickNumber ((n/x/10)% 10);
    delayMicroseconds (del);
    clearLEDs ();
    pickDigit (4);
    pickNumber (n / x% 10);
    delayMicroseconds (del);
    n + +;
    if (digitalRead (13) == HIGH) { n = 0; }
}
void loop ()
{
    digitalWrite (d1, LOW);
    digitalWrite (d2, LOW);

```

```

digitalWrite (d3, LOW);
digitalWrite (d4, LOW);
switch (x) {
    case 1:
        digitalWrite (d1, HIGH);
        break;
    case 2:
        digitalWrite (d2, HIGH);
        break;
    case 3:
        digitalWrite (d3, HIGH);
        break;
    default: digitalWrite (d4, HIGH); break;
}
}
void pickNumber (int x) //          pickNumber (x),
{
    switch (x) {
        default:
            zero ();
            break;
        case 1:
            one ();
            break;
        case 2:
            two ();
            break;
        case 3:
            three ();
            break;
        case 4:
            four ();
            break;
        case 5:
            five ();
            break;
        case 6:
            six ();
            break;
        case 7:
            seven ();
            break;
        case 8:
            eight ();
            break;
        case 9:
            nine ();
            break;
    }
}

```

```

    }
}
void clearLEDs () //
{
    digitalWrite (a, HIGH);
    digitalWrite (b, HIGH);
    digitalWrite (c, HIGH);
    digitalWrite (d, HIGH);
    digitalWrite (e, HIGH);
    digitalWrite (f, HIGH);
    digitalWrite (g, HIGH);
    digitalWrite (p, HIGH);
}
void one () //          1:00
{
    digitalWrite (a, HIGH);
    digitalWrite (b, LOW);
    digitalWrite (c, LOW);
    digitalWrite (d, HIGH);
    digitalWrite (e, HIGH);
    digitalWrite (f, HIGH);
    digitalWrite (g, HIGH);
}
void two () //          2:00
{
    digitalWrite (a, LOW);
    digitalWrite (b, LOW);
    digitalWrite (c, HIGH);
    digitalWrite (d, LOW);
    digitalWrite (e, LOW);
    digitalWrite (f, HIGH);
    digitalWrite (g, LOW);
}
void three () //          3:00

```

```
{
    digitalWrite (a, LOW);
    digitalWrite (b, LOW);
    digitalWrite (c, LOW);
    digitalWrite (d, LOW);
    digitalWrite (e, HIGH);
    digitalWrite (f, HIGH);
    digitalWrite (g, LOW);
}
void four () //          4:00
{
    digitalWrite (a, HIGH);
    digitalWrite (b, LOW);
    digitalWrite (c, LOW);
    digitalWrite (d, HIGH);
    digitalWrite (e, HIGH);
    digitalWrite (f, LOW);
    digitalWrite (g, LOW);
}
void five () //          5:00
{
    digitalWrite (a, LOW);
    digitalWrite (b, HIGH);
    digitalWrite (c, LOW);
    digitalWrite (d, LOW);
    digitalWrite (e, HIGH);
    digitalWrite (f, LOW);
    digitalWrite (g, LOW);
}
void six () //          6:00
{
    digitalWrite (a, LOW);
    digitalWrite (b, HIGH);
    digitalWrite (c, LOW);
    digitalWrite (d, LOW);
    digitalWrite (e, LOW);
    digitalWrite (f, LOW);
    digitalWrite (g, LOW);
}
void seven () //        7:00
{
    digitalWrite (a, LOW);
    digitalWrite (b, LOW);
    digitalWrite (c, LOW);
    digitalWrite (d, HIGH);
    digitalWrite (e, HIGH);
    digitalWrite (f, HIGH);
    digitalWrite (g, HIGH);
}
```

```

}
void eight () //          8:00
{
    digitalWrite (a, LOW);
    digitalWrite (b, LOW);
    digitalWrite (c, LOW);
    digitalWrite (d, LOW);
    digitalWrite (e, LOW);
    digitalWrite (f, LOW);
    digitalWrite (g, LOW);
}
void nine () //          9:00
{
    digitalWrite (a, LOW);
    digitalWrite (b, LOW);
    digitalWrite (c, LOW);
    digitalWrite (d, LOW);
    digitalWrite (e, HIGH);
    digitalWrite (f, LOW);
    digitalWrite (g, LOW);
}

```

Á Á ¨ ¨ Á Á Á Á È

Á Á K

```

const int row2 = 3; // 14
const int row3 = 4; // 8
const int row4 = 5; // 12
const int row5 = 17; // 1
const int row6 = 16; // 7
const int row7 = 15; // 2
const int col1 = 6; // 13
const int col2 = 7; // 3
const int col3 = 8; // 4
const int col4 = 9; // 10
const int col5 = 10; // 6
const int col6 = 11; // 11
const int col7 = 12; // 15
const int col8 = 13; // 16
void setup ()
{
    int i = 0; for (i = 2; i < 18; i++)

```

```

{
    pinMode (i, OUTPUT);
}

pinMode (row5, OUTPUT);
pinMode (row6, OUTPUT);
pinMode (row7, OUTPUT);
pinMode (row8, OUTPUT);
for (i = 2; i <18; i + +)
{
    digitalWrite (i, LOW);
}
digitalWrite (row5, LOW);
digitalWrite (row6, LOW);
digitalWrite (row7, LOW);
digitalWrite (row8, LOW);
}
void loop ()
{
    digitalWrite (row1, HIGH);
    digitalWrite (row2, LOW);
    digitalWrite (row3, LOW);
    digitalWrite (row4, LOW);
    digitalWrite (row5, LOW);
    digitalWrite (row6, LOW);
    digitalWrite (row7, LOW);
    digitalWrite (row8, LOW);
    digitalWrite (col1, LOW);
    digitalWrite (col2, HIGH);
    digitalWrite (col3, HIGH);
    digitalWrite (col4, HIGH);
    digitalWrite (col5, HIGH);
    digitalWrite (col6, HIGH);
    digitalWrite (col7, HIGH);
    digitalWrite (col8, HIGH);
    delay (1000); //
    for (i = 2; i <18; i + +)
    {
        digitalWrite (i, LOW);
    }
    delay (1000);
}

```

Define display_array_size 8 // Ascii 8x8 dot font

Define data_null 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 // null char

```

# Define data_ascii_A 0x02, 0x0C, 0x18, 0x68, 0x68, 0x18, 0x0C, 0x02 / * "A", 0 * /
/ **
** "A"
# Define A { //
    {0, 0, 0, 0, 0, 0, 1, 0}, // 0x02
    {0, 0, 0, 0, 1, 1, 0, 0}, // 0x0C
    {0, 0, 0, 1, 1, 0, 0, 0}, // 0x18
    {0, 1, 1, 0, 1, 0, 0, 0}, // 0x68
    {0, 1, 1, 0, 1, 0, 0, 0}, // 0x68
    {0, 0, 0, 1, 1, 0, 0, 0}, // 0x18
    {0, 0, 0, 0, 1, 1, 0, 0}, // 0x0C
    {0, 0, 0, 0, 0, 0, 1, 0} // 0x02
}
** /

# Define data_ascii_B 0x00, 0x7E, 0x52, 0x52, 0x52, 0x52, 0x2C, 0x00 / * "B", 1 * /
# Define data_ascii_C 0x00, 0x3C, 0x66, 0x42, 0x42, 0x42, 0x2C, 0x00 / * "C", 2 * /
# Define data_ascii_D 0x00, 0x7E, 0x42, 0x42, 0x42, 0x66, 0x3C, 0x00 / * "D", 3 * /
# Define data_ascii_E 0x00, 0x7E, 0x52, 0x52, 0x52, 0x52, 0x52, 0x42 / * "E", 4 * /
# Define data_ascii_F 0x00, 0x7E, 0x50, 0x50, 0x50, 0x50, 0x50, 0x40 / * "F", 5 * /
# Define data_ascii_G 0x00, 0x3C, 0x66, 0x42, 0x42, 0x52, 0x16, 0x1E / * "G", 6 * /
# Define data_ascii_H 0x00, 0x7E, 0x10, 0x10, 0x10, 0x10, 0x7E, 0x00 / * "H", 7 * /
# Define data_ascii_I 0x00, 0x00, 0x00, 0x7E, 0x00, 0x00, 0x00, 0x00 / * "I", 8 * /
// Display array
byte data_ascii [] [display_array_size] = {
    data_null,
    data_ascii_A,
    data_ascii_B,
    data_ascii_C,
    data_ascii_D,
    data_ascii_E,
    data_ascii_F,
    data_ascii_G,
    data_ascii_H,
    data_ascii_I,
};

// The pin to control ROW
const int row1 = 2; // the number of the row pin 24
const int row2 = 3; // the number of the row pin 23
const int row3 = 4; // the number of the row pin 22
const int row4 = 5; // the number of the row pin 21
const int row5 = 17; // the number of the row pin 4
const int row6 = 16; // the number of the row pin 3
const int row7 = 15; // the number of the row pin 2
const int row8 = 14; // the number of the row pin 1
//
const int col1 = 6; // the number of the col pin 20
const int col2 = 7; // the number of the col pin 19
const int col3 = 8; // the number of the col pin 18

```

```
const int col4 = 9; // the number of the col pin 17
const int col5 = 10; // the number of the col pin 16
const int col6 = 11; // the number of the col pin 15
const int col7 = 12; // the number of the col pin 14
const int col8 = 13; // the number of the col pin 13
```

```
void displayNum (byte rowNum, int colNum)
{
    int j;
    byte temp = rowNum;
    for (j = 2; j <6; j ++ )
    {
        digitalWrite (j, LOW);
    }
    digitalWrite (row5, LOW);
    digitalWrite (row6, LOW);
    digitalWrite (row7, LOW);
    digitalWrite (row8, LOW);
    for (j = 6; j <14; j ++ )
    {
        digitalWrite (j, HIGH);
    }
    switch (colNum)
    {
        case 1:
            digitalWrite (col1, LOW);
            break;
        case 2:
            digitalWrite (col2, LOW);
            break;
        case 3:
            digitalWrite (col3, LOW);
            break;
        case 4:
            digitalWrite (col4, LOW);
            break;
        case 5:
            digitalWrite (col5, LOW);
            break;
        case 6:
            digitalWrite (col6, LOW);
            break;
        case 7:
            digitalWrite (col7, LOW);
            break;
        case 8:
            digitalWrite (col8, LOW);
            break;
```

```

        default: break;
    }

    for (j = 1; j < 9; j++)
    {
        temp = (0x80) & (temp);
        if (temp == 0)
        {
            temp = rowNum << j; continue;
        }
        switch (j)
        {
            case 1:
                digitalWrite (row1, HIGH);
                break;
            case 2:
                digitalWrite (row2, HIGH);
                break;
            case 3:
                digitalWrite (row3, HIGH);
                break;
            case 4:
                digitalWrite (row4, HIGH);
                break;
            case 5:
                digitalWrite (row5, HIGH);
                break;
            case 6:
                digitalWrite (row6, HIGH);
                break;
            case 7:
                digitalWrite (row7, HIGH);
                break;
            case 8:
                digitalWrite (row8, HIGH);
                break;
            default:
                break;
        }
        temp = rowNum << j;
    }
}

void setup ()
{
    int i = 0; for (i = 2; i < 18; i++)
    {
        pinMode (i, OUTPUT);
    }
}

```

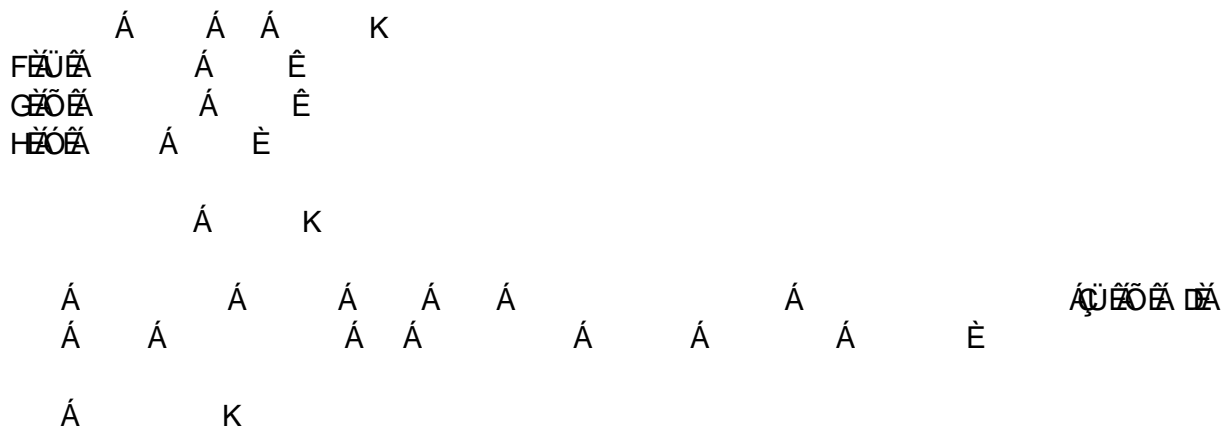
```

    }
    for (i = 2; i <18; i++)
    {
        digitalWrite (i, LOW);
    }
}
void loop ()
{
    int t1;
    int l;
    int arrage;
    for (arrage = 0; arrage <10; arrage++)
    {
        for (l = 0; l <512; l++)
        {
            for (t1 = 0; t1 <8; t1++)
            {
                displayNum (data_ascii [arrage] [t1], (t1 +1));
            }
        }
    }
}

```

Robstore.ru

21 - LED RGB



```
int ledPin = 13; // LED is connected to digital pin 13
int redPin = 11; // R petal on RGB LED module connected to digital pin 11
int greenPin = 9; // G petal on RGB LED module connected to digital pin 9
int bluePin = 10; // B petal on RGB LED module connected to digital pin 10
void setup ()
{
    pinMode (ledPin, OUTPUT); // output
    pinMode (redPin, OUTPUT); // sets the redPin to be an output
    pinMode (greenPin, OUTPUT); // sets the greenPin to be an output
    pinMode (bluePin, OUTPUT); // sets the bluePin to be an output
}
void loop () // run over and over again
{
    // Basic colors:
    color (255, 0, 0); // turn the RGB LED red
    delay (1000); // delay for 1 second
    color (0,255, 0); // turn the RGB LED green
    delay (1000); // delay for 1 second
    color (0, 0, 255); // turn the RGB LED blue
    delay (1000); // delay for 1 second
    // Example blended colors:
    color (255,255,0); // turn the RGB LED yellow
    delay (1000); // delay for 1 second
    color (255,255,255); // turn the RGB LED white
    delay (1000); // delay for 1 second
    color (128,0,255); // turn the RGB LED purple
    delay (1000); // delay for 1 second
    color (0,0,0); // turn the RGB LED off
    delay (1000); // delay for 1 second
}
void color (unsigned char red, unsigned char green, unsigned char blue) // the color
generating function
{
    analogWrite (redPin, 255-red);
```

```
    analogWrite (bluePin, 255-blue);  
    analogWrite (greenPin, 255-green);  
}
```

Robstore.ru


```

void setup ()
{
    pinMode (data, OUTPUT);
    pinMode (clock, OUTPUT);
    pinMode (latch, OUTPUT);
}
void loop ()
{
    int delayTime = 100;
    for (int i = 0; i <256; i ++ )
    {
        updateLEDs (i);
        delay (delayTime);
    }
}
void updateLEDs (int value)
{
    digitalWrite (latch, LOW);
    shiftOut (data, clock, MSBFIRST, value);
    digitalWrite (latch, HIGH);
}
void updateLEDsLong (int value)
{
    digitalWrite (latch, LOW);
    for (int i = 0; i <8; i ++ )
    {
        int bit = value & B10000000;
        value = value << 1;
        if (bit == 128)
        {
            digitalWrite (data, HIGH);
        }
        else
        {
            digitalWrite (data, LOW);
        }
        digitalWrite (clock, HIGH);
        delay (1);
        digitalWrite (clock, LOW);
    }
    digitalWrite (latch, HIGH);
}
int bits [] = {B00000001, B00000010, B00000100, B00001000, B00010000, B00100000,
B01000000, B10000000};
int masks [] = {B11111110, B11111101, B11111011, B11110111, B11101111, B11011111,
B10111111, B01111111};
void changeLED (int led, int state)
{

```

```
ledState = ledState & masks [led];  
if (state == ON)  
{  
    ledState = ledState | bits [led];  
}  
updateLEDs (ledState);  
}
```

Á Á Á Á Á Á Á Á Ě

'&' '-' ' (("

Á K

Á K

1. Á Á Á I K Á È
 2. Á Á Á È
 3. Á Á Á Á Á
- Á Á Á Á Á [K

Robstore.ru

```

    Á      Á      Á      Á      Á      Á      Á      Á      Á
  # Include const byte ROWS = 4; // . 4
  const byte COLS = 4; // 4
  char keys [ROWS] [COLS] = { {'1 ', '2', '3 ', 'A '}, {'4 ', '5', '6 ', 'B '}, {'7 ', '8', '9 ', 'C '}, {'*', '0 ', ' ' #
  ', 'D '}}; //
  à^ ¢ Á[ , Úā • ÆÜUY ùÁMÇÊ Ê ðÁÁ      Á      ÁÁ      Á      Á
  byte colPins [COLS] = {6,7,8,9};
  ÁÁ      Á      ÁS^~] æ
  keypad = Keypad (makeKeymap (keys), rowPins, colPins, ROWS, COLS);
  void setup () {
    Serial.begin (9600);
  }
  void loop ()
  {
    char key = keypad.getKey ();
    if (key!= NO_KEY)
    {
      Serial.println (key);
    }
  }
  Á      Á      ÆÁ      ÁÁ      Æ      Á      ÁÁ
  Á      È
  Á      ÁÁ      Á      Á      Á I Æ      Á      Á      Á
  Á      ÁHÁ      Á      Á      È
  # Include const byte ROWS = 4; // . 4
  const byte COLS = 4; // 4
  char keys [ROWS] [COLS] = { {'1 ', '2', '3 ', 'A '}, {'4 ', '5', '6 ', 'B '}, {'7 ', '8', '9 ', 'C '}, {'*', '0 ', ' ' #
  ', 'D '}}; //
  à^ ¢ Á[ , Úā • ÆÜUY ùÁMÇÊ Ê ðÁÁ      Á      ÁÁ      Á      Á
  byte colPins [COLS] = {6,7,8,9};
  ÁÁ      Á      ÁS^~] æ
  keypad = Keypad (makeKeymap (keys), rowPins, colPins, ROWS, COLS);

```

```
byte ledPin = 13;
boolean blink = false;
void setup ()
{
    Serial.begin (9600);
    pinMode (ledPin, OUTPUT); // sets the digital pin as output
    digitalWrite (ledPin, HIGH); // sets the LED on
    keypad.addEventListener (keypadEvent); // add an event listener for this keypad
}
void loop ()
{
    char key = keypad.getKey ();
    if (key != NO_KEY) { Serial.println (key); }
    if (blink) {
        digitalWrite (ledPin,! digitalRead (ledPin));
        delay (100);
    }
}
// Take care of some special events
void keypadEvent (KeypadEvent key)
{
    switch (keypad.getState ())
    {
        case PRESSED:
            switch (key)
            {
                case '#':
                    digitalWrite (ledPin,! digitalRead (ledPin));
                    break;
                case '*':
                    digitalWrite (ledPin,! digitalRead (ledPin));
                    break;
            }
            break;
        case RELEASED:
            switch (key)
            {
                case '*':
                    digitalWrite (ledPin,! digitalRead (ledPin));
                    blink = false;
                    break;
            }
            break;
        case HOLD:
            switch (key)
            {
                case '*':
                    blink = true;
            }
            break;
    }
}
```

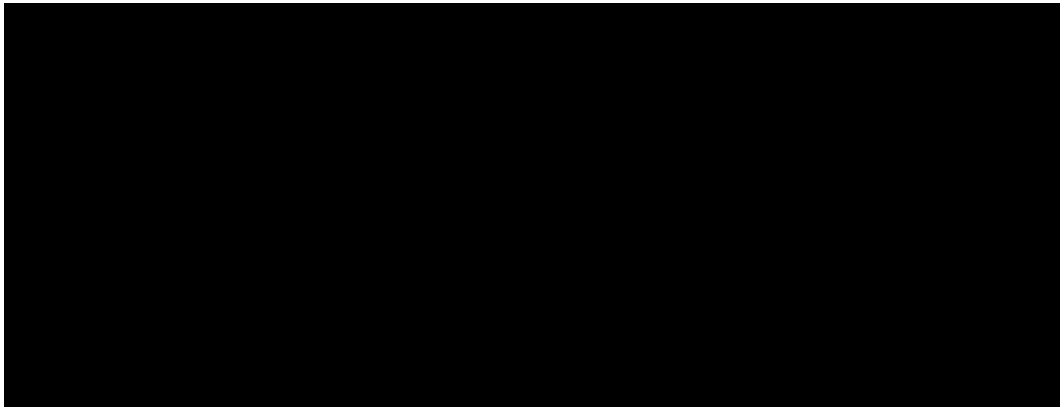
```

                                break;
                            }
                        break;
                    }
                }
            }
        }
    }
}

```

·8 G% \$& ·5fXi]bc"

ÀÙFHÉGÁ Á Á Á É É É É É É É
 Á Á Á Á Á Á
 ÀÉà~ q[Á Á Á Á È



Á Á ~~Á~~

<https://datasheets.maximintegrated.com/en/ds/DS1302.pdf>

| Á | KA |
|------------------------------------|----|
| CE (DS1302 pin5) -> Arduino D5 | |
| IO (DS1302 pin6) -> Arduino D6 | |
| SCLK (DS1302 pin7) -> Arduino D7 | |
| Vcc2 (DS1302 pin1) -> Arduino +5 v | |
| GND (DS1302 pin4) -> Arduino GND | |

•

•

```
# Include <stdio.h>
# Include <DS1302.h>
```

```

/* Interface Definition
CE (DS1302 pin5) -> Arduino D5
IO (DS1302 pin6) -> Arduino D6
SCLK (DS1302 pin7) -> Arduino D7 */

```

```
uint8_t CE_PIN = 5;
uint8_t IO_PIN = 6;
uint8_t SCLK_PIN = 7;
```

```
char buf [50];
char day [10];
```

```
/*  
String comdata = "";  
int numdata [7] = {0}, j = 0, mark = 0;
```

0, mark = 0;

```

/*                      DS1302 */
DS1302 rtc (CE_PIN, IO_PIN, SCLK_PIN);

void print_time () {
    Time t = rtc.time ();
    memset (day, 0, sizeof (day));
    switch (t.day)
    {
        case 1:
            strcpy (day, "Sunday");
            break;
        case 2:
            strcpy (day, "Monday");
            break;
        case 3:
            strcpy (day, "Tuesday");
            break;
        case 4:
            strcpy (day, "Wednesday");
            break;
        case 5:
            strcpy (day, "Thursday");
            break;
        case 6:
            strcpy (day, "Friday");
            break;
        case 7:
            strcpy (day, "Saturday");
            break;
    }
    snprintf (buf, sizeof (buf), "% s% 04d-% 02d-% 02d% 02d:% 02d:% 02d", day, t.yr, t.mon,
    t.date, t.hr, t. min, t.sec);
    Serial.println (buf);
}

void setup ()
{
    Serial.begin (9600);
    rtc.write_protect (false);
    rtc.halt (false);
}

void loop ()
{
    delay (1000);
}

```



```

// Set the LED pin as an output
pinMode(LED_PIN, OUTPUT);

// Turn the LED on (HIGH)
digitalWrite(LED_PIN, HIGH);

// Turn the LED off (LOW)
digitalWrite(LED_PIN, LOW);

```

```

void setup ()
{
    pinMode (led, OUTPUT);
    digitalWrite (led, LOW);
}

void loop ()
{
    if (val > 700)
    {
        digitalWrite (led, HIGH); // LED on
    }
    else
    {
        digitalWrite (led, LOW); // LED off
    }
    data = val;
}

```

}

Á

Á

Á

Á

Á

Á

Á

 \hat{A}

Á

Á

Á

Á

Á

Á

Á

Robstore.ru

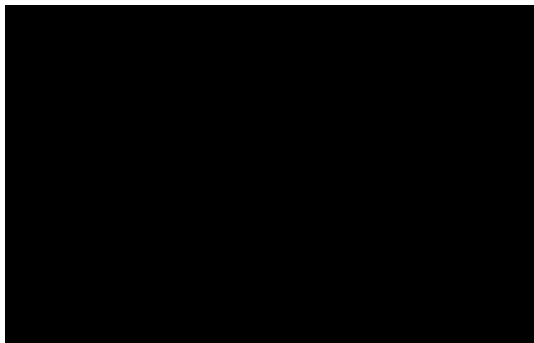
'&* '-

'8 < Hº/%'

É Á Ě Á Á Ě Á Ě Á -
Á Á Á Á Ě Á É Á Á Ě Ě
Á Ě Á - Á Á Á
Á Ě Á Á Á Á Á É

1. Á K
2. Á Ě Á Ě Á
3. Á Ě
4. Ě
5. Á Á Ě Á Á Ě Á Ě Á
6. Á Ě Á Ě Á Ě Á Ě Á Ě Á 50
Á Á Á Á Ě Á Ě Á

Á K



1. Arduino K Ě
2. ÖPVÁFÁ

Á K

```
int DHpin = 8;
byte dat [5];
byte read_data ()
{
    byte data;
    for (int i = 0; i < 8; i ++ )
    {
        if (digitalRead (DHpin) == LOW)
        {
            while (digitalRead (DHpin) == LOW); // 50us;
            á^|æ T æ[ • ^ & ] á • Á Ě Á
            Á Á Á
        }
    }
}
```

```

        if (digitalRead (DHpin) == HIGH)
            data |= (1 << (7 - i));
        i++;
    }
    return data;
}

void start_test ()
{
    digitalWrite (DHpin, HIGH);
    pinMode (DHpin, INPUT);
    while (digitalRead (DHpin) == HIGH);
    delayMicroseconds (80); // DHT11 80us;
    if (digitalRead (DHpin) == LOW);
    delayMicroseconds (80); // DHT11 80us;
    dat[i] = read_data ();
    pinMode (DHpin, OUTPUT);
}

void setup ()
{
    Serial.begin (9600);
    pinMode (DHpin, OUTPUT);
}

void loop ()
{
    start_test ();
    Serial.print ("Current humidity =");
    Serial.print ('.');
    Serial.println ("%");
    Serial.print ("Current temperature =");
    Serial.print ('.');
    Serial.println ('C'); delay (700);
}

```

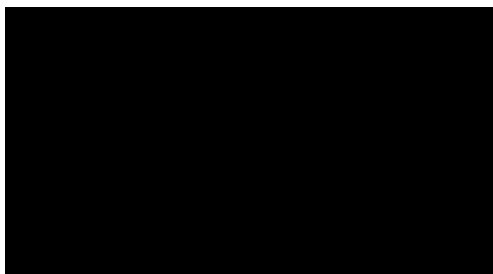
Á Á Ê Á Á Ê Á Á Á Á Á Á

Robstore.ru

Á Á ÁÁ ÁÁ Á Á Á Á Á
Ě Ě Ě Ě Ě Ě Ě Ě Ě Ě

Robstore.ru

"

[illegible]

- Á K
1. XÔÔÁ ÁÁ Á Á È

ÁÇ ÁĖ Ĭ DK
Á
Á Á È

2. $\tilde{O}P\ddot{O}A \ A \quad A \ \tilde{O}P\ddot{O}A \quad \overline{\overline{\overline{\overline{A}}}} \quad E$
3. $\overline{O}A \ A \quad A \ A \quad A \quad \text{Đ} \quad A \quad \text{Ç}A \quad A \quad \text{ÁÁ}$
- 4).

1. $\overline{O}A \sim \overline{a} \ [\ A \quad K \quad \overline{K}A \ E$
2. $A \quad \overline{K}A \ E$
3. $\tilde{S}\ddot{O}A \quad \overline{K}A \ E$
4. $H\text{€} \ A \quad \overline{K}A \ E$

$A \quad K$

$\overline{a} \ \overline{a}^{\wedge} | \overline{a} \ \overline{A} \ \overline{K} \ \overline{A} \quad A \quad L$

void setup ()

{

pinMode $\overline{C}^{\wedge} | \overline{a} \ \overline{A} \ \overline{U} \ \overline{W} \ \overline{U} \ \overline{W} \ \overline{D} \ \overline{A} \ A \quad A \ \overline{U} \ \overline{W} \ \overline{U} \ \overline{W}$

}

void loop ()

{

digitalWrite (relay, HIGH); //

delay (1000);

digitalWrite (relay, LOW); //

delay (1000);

}

$A \quad A \quad A \ A \quad \tilde{S}\ddot{O}A \quad A \quad \overline{K}A \ \overline{A}A \quad E$

Robstore.ru

'&,'- '=87'@78%\$&"

ŠÔÖÁ ŠÔÖÁ Á Á ĚÁ Á Á Á Á Á Á Á Á
Á Á Á Á Á Á Á Á Đ È Á Á Á Á
ÁÔÖŠÔÖÏ €GÁ Á Á Á Á Á Á Á Á Á Á
Á Á Á È ĚÁ Á Á Á

Á K
1. KĚ -6V
2. KÁ FĚ Á Ć DĚ JÁ Ć DĚ Í È Á Ć D
3. KÁ
4. Á I2C
5. Á Á Á Á
Á K

Á K
1. Œâ~ā[Á KĚÁ È
2. ŒÔŠÔÖÏ €GÁ KĚÁ È
Á Á Á Á Á Á Á
:
Ä[à•q !^È~

#include <wire.h>
#include <LiquidCristal_I2C.h>

Robstore.ru

[illegible]

- 1.
- 2.
- 3.
4. Self Positioning Torque: ~34mN
5. Friction Torque: 60~120mN
6. Pull-in Torque: ~30mN
- 7.

[illegible]

Á Á Á Á Á

/ *

* Stepper motor rotating follower potentiometer

* (Or other sensors) using the input analog port number 0

* Use the arduino IDE comes Stepper.h library files

*/

Include <Stepper.h>

~~#~~ ~~A~~ ~~A~~

Define STEPS 100

~~#~~ ~~A~~ ~~A~~ ~~A~~

Stepper stepper (STEPS, 8, 9, 10, 11);

//

int previous = 0;

void setup ()

{

~~#~~ ~~A~~ ~~A~~ ~~A~~ ~~A~~ ~~A~~

stepper.setSpeed (90);

}

void loop ()

{

~~#~~ ~~A~~ ~~A~~

int val = analogRead (0);

// ~~A~~ ~~A~~ ~~A~~ ~~A~~ ~~A~~ ~~A~~

stepper.step (val - previous);

//

previous = val;

}

Robstore.ru

· \$-

11

•

11

[illegible]

ÀŒŒÁ ÁÁ Á Á Á Á Á
 È Á Á Á Á ÁÁ ÁŒÁ È

[illegible][illegible]

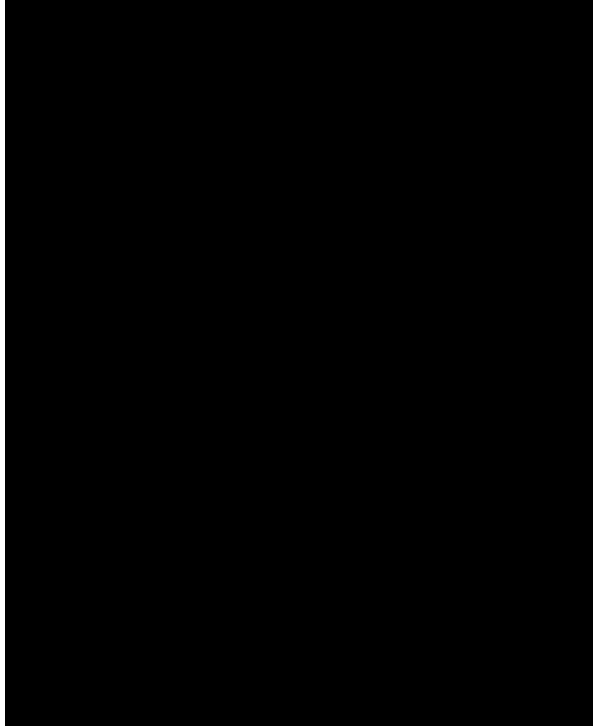
[illegible]

1. Á K
2. Á K È

Á Á Á Á Á Á Á Æ Á
 Á Á Á Á Á Á Á

Æ È Æ Á Å å ä | Á È

Y, Key.

[illegible]

```
int sensorPin = 5;
int value = 0;
void setup ()
{
    pinMode (7, OUTPUT);
    Serial.begin (9600);
}
void loop ()
{
    value = analogRead (0);
    Serial.print ("X:");
```

```
PUT);
0);

head (0);
```

```
Serial.print (value, DEC);  
value = analogRead (1);  
Serial.print ("| Y:");  
Serial.print (value, DEC);  
value = digitalRead (7);  
Serial.print ("| Z:");  
Serial.println (value, DEC);  
delay (100);  
}
```

Á Á Á Á Á Á Á Á

Robstore.ru

100

```
cv::decode()
```

```

// Decode_results * results = (decode_results *) v
void dump (decode_results * results) {
    int count = results-> rawlen;
    if (results-> decode_type == UNKNOWN)
    {
        Serial.println ("Could not decode message");
    }
    else
    {
        if (results-> decode_type == NEC)
        {
            Serial.print ("Decoded NEC:");
        }
        else if (results-> decode_type == SONY)
        {
            Serial.print ("Decoded SONY:");
        }
        else if (results-> decode_type == RC5)
        {
            Serial.print ("Decoded RC5:");
        }
        else if (results-> decode_type == RC6)
        {
            Serial.print ("Decoded RC6:");
        }
        Serial.print (results-> value, HEX);
        Serial.print ("(");
        Serial.print (results-> bits, DEC);
        Serial.println ("bits)");
    }
    Serial.print ("Raw (");
    Serial.print (count, DEC);
    Serial.print ("):");
    for (int i = 0; i < count; i++)
    {
        if ((i% 2) == 1) {
            Serial.print (results-> rawbuf [i] * USECPERTICK, DEC);
        }
        else
        {
            Serial.print (- (int) results-> rawbuf [i] * USECPERTICK, DEC);
        }
        Serial.print ("");
    }
    Serial.println ("");
}

void setup ()
{

```

```

pinMode (RECV_PIN, INPUT);
pinMode (LED1, OUTPUT);
pinMode (LED2, OUTPUT);
pinMode (LED3, OUTPUT);
pinMode (LED4, OUTPUT);
pinMode (LED5, OUTPUT);
pinMode (LED6, OUTPUT);
pinMode (13, OUTPUT);
Serial.begin (9600);
irrecv.enableIRIn (); //
}
int on = 0;
unsigned long last = millis ();
void loop ()
{
  if (irrecv.decode (& results))
  {
    //      Á  Á  Æ  Á      Á  Á  Á  Á  Á
    on -      Æ      Á
    if (millis () - last > 250)
    {
      on = ! on;
      // digitalWrite (8, on? HIGH: LOW);
      digitalWrite (13, on? HIGH: LOW);
      dump (& results);
    }
    if (results.value == on1)
      digitalWrite (LED1, HIGH);
    if (results.value == off1)
      digitalWrite (LED1, LOW);
    if (results.value == on2)
      digitalWrite (LED2, HIGH);
    if (results.value == off2)
      digitalWrite (LED2, LOW);
    if (results.value == on3)
      digitalWrite (LED3, HIGH);
    if (results.value == off3)
      digitalWrite (LED3, LOW);
    if (results.value == on4)
      digitalWrite (LED4, HIGH);
    if (results.value == off4)
      digitalWrite (LED4, LOW);
    if (results.value == on5)
      digitalWrite (LED5, HIGH);
    if (results.value == off5)
      digitalWrite (LED5, LOW);
    if (results.value == on6)
      digitalWrite (LED6, HIGH);
  }
}

```

```

    if (results.value == off6)
        digitalWrite (LED6, LOW);
    last = millis ();
    if (millis () - last > 1000)
    {
        digitalWrite (LED6, HIGH);
        delay (500);
        digitalWrite (LED6, LOW);
        delay (500);
    }
}

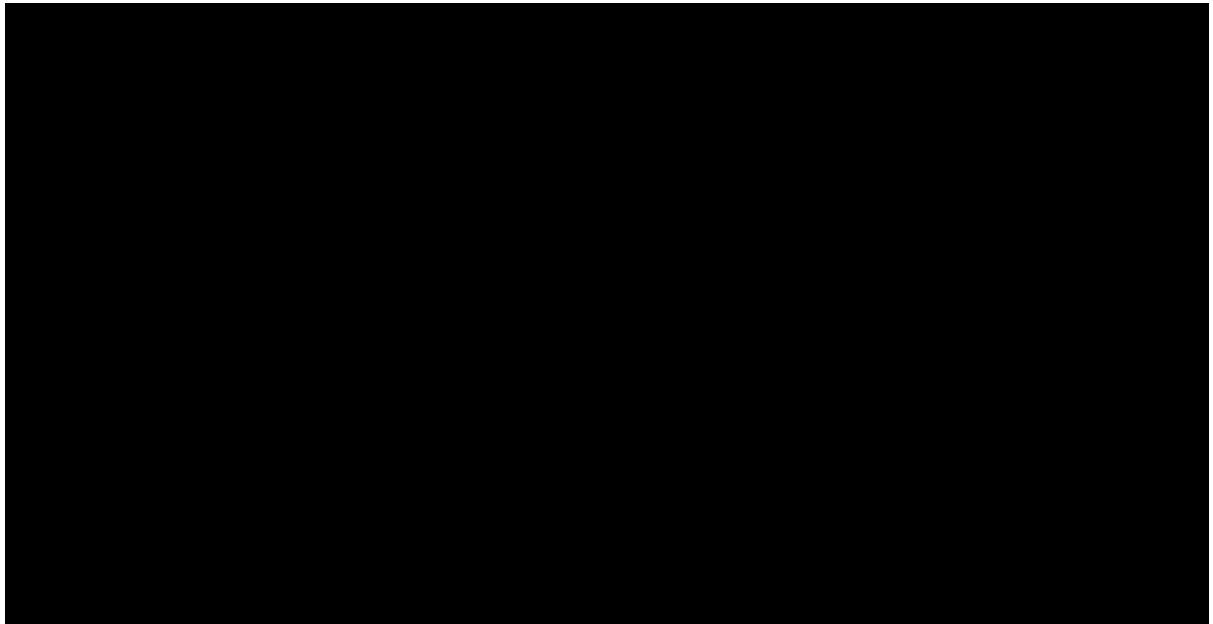
```

"F: 8"

ÜÁ Á Á Á ÄÖËÄ Á Á

È

Á K



Á ÄÖËÄ ÄÖËÄ Á Á Á Á - ÄÖËÄ

Á Á Á Á Á Á Á Á Á Á

Á Á Á Á Á Á Á Á Á Á

Á ÄÖËÄ Á ÄÖËÄ Á Á Á Á Á Á

Á Á Á Á Á Á Á Á Á Á

```

# Include <SPI.h>
# Define uchar unsigned char
# Define uint unsigned int
#define MAX_LEN 16
# Define MAX_LEN 16
  
```

Robstore.ru

```
//
const int chipSelectPin = 10; // Arduino uno
const int chipSelectPin = 53; // Arduino mega 2560,1280
const int NRSTPD = 5;
// #define CC_A
// #define CC_A
// #define CC_A
// Define PCD_AUTHENT 0x0E //
// Define PCD_RECEIVE 0x08 //
// Define PCD_TRANSMIT 0x04 //
// #define CC_A
// #define CC_A
// Define PCD_RESETPHASE 0x0F // Reset
// Define PCD_CALCCRC 0x03 // CRC
// #define CC_A
// #define CC_A
// #define CC_A
// #define CC_A
// #define CC_A
// Define PICC_ANTICOLL 0x93 // anti-collision
// Define PICC_SEIECTTAG 0x93 // election card
// Define PICC_AUTHENT1A 0x60 // authentication key A
// Define PICC_AUTHENT1B 0x61 // authentication key B
// Define PICC_READ 0x30 //
// Define PICC_WRITE 0xA0 //
// Define PICC_DECREMENT 0xC0
// Define PICC_INCREMENT 0xC1
// #define CC_A
// #define CC_A
// #define CC_A
// #define CC_A
// Define PICC_HALT 0x50 // Sleep
// #define CC_A
// Define MI_OK 0
// Define MI_NOTAGERR 1
// Define MI_ERR 2
// ----- MFRC522 -----
// Page 0:
// Define Reserved00 0x00
// Define CommandReg 0x01
// Define CommEnReg 0x02
// Define DivlEnReg 0x03
// Define CommIrqReg 0x04
// Define DivlIrqReg 0x05
// Define ErrorReg 0x06
// Define Status1Reg 0x07
// Define Status2Reg 0x08
// Define FIFODataReg 0x09
// Define FIFOLevelReg 0x0A
// Define WaterLevelReg 0x0B
// Define ControlReg 0x0C
// Define BitFramingReg 0x0D
// Define CollReg 0x0E
// Define Reserved01 0x0F
```

```
// Page 1: Command
# Define Reserved10 0x10
# Define ModeReg 0x11
# Define TxModeReg 0x12
# Define RxModeReg 0x13
# Define TxControlReg 0x14
# Define TxAutoReg 0x15
# Define TxSelReg 0x16
# Define RxSelReg 0x17
# Define RxThresholdReg 0x18
# Define DemodReg 0x19
# Define Reserved11 0x1A
# Define Reserved12 0x1B
# Define MifareReg 0x1C
# Define Reserved13 0x1D
# Define Reserved14 0x1E
# Define SerialSpeedReg 0x1F
// Page 2: CFG
# Define Reserved20 0x20
# Define CRCResultRegM 0x21
# Define CRCResultRegL 0x22
# Define Reserved21 0x23
# Define ModWidthReg 0x24
# Define Reserved22 0x25
# Define RFCfgReg 0x26
# Define GsNReg 0x27
# Define CWGsPReg 0x28
# Define ModGsPReg 0x29
# Define TModeReg 0x2A
# Define TPrescalerReg 0x2B
# Define TReloadRegH 0x2C
# Define TReloadRegL 0x2D
# Define TCounterValueRegH 0x2E
# Define TCounterValueRegL 0x2F
// Page 3: TestRegister
# Define Reserved30 0x30
# Define TestSel1Reg 0x31
# Define TestSel2Reg 0x32
# Define TestPinEnReg 0x33
# Define TestPinValueReg 0x34
# Define TestBusReg 0x35
# Define AutoTestReg 0x36
# Define VersionReg 0x37
# Define AnalogTestReg 0x38
# Define TestDAC1Reg 0x39
# Define TestDAC2Reg 0x3A
# Define TestADCReg 0x3B
# Define Reserved31 0x3C
```

```

# Define Reserved32 0x3D
# Define Reserved33 0x3E
# Define Reserved34 0x3F
// -----
uchar serNum [5];
uchar writeDate [16] = {'T', 'e', 'n', 'g', ' ', 'B', 'o', 0, 0, 0, 0, 0, 0, 0, 0, 0};
// Sector A password, 16
uchar sectorKeyA [16] [16] = {{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
};
uchar sectorNewKeyA [16] [16] = {{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
0xff, 0x07, 0x80, 0x69, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
0xff, 0x07, 0x80, 0x69, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF},
};
void setup () {
    Serial.begin (9600); // RFID reader SOUT pin connected to Serial
    RX pin at 2400bps
    // Start SPI library:
    SPI.begin ();
    pinMode (chipSelectPin, OUTPUT); // OUTPUT
    digitalWrite (chipSelectPin, LOW); // RFID
    pinMode (NRSTPD, OUTPUT); // OUTPUT
    digitalWrite (NRSTPD, HIGH);
    MFRC522_Init ();
}
void loop ()
{
    uchar i, tmp;
    uchar status;
    uchar str [MAX_LEN];
    uchar RC_size;
    uchar blockAddr; // 0 63
    //
    status = MFRC522_Request (PICC_REQIDL, str);
    if (status == MI_OK)
    {
    }
    // Anti-collision, 4
    status = MFRC522_Anticoll (str);
    memcpy (serNum, str, 5);
    if (status == MI_OK)
    {
        Serial.println ("The card's number is:");
        Serial.print (serNum [0], BIN);
    }
}

```

```

        Serial.print (serNum [1], BIN);
        Serial.print (serNum [2], BIN);
        Serial.print (serNum [3], BIN);
        Serial.print (serNum [4], BIN);
        Serial.println ("");
    }
    //
    RC_size = MFRC522_SelectTag (serNum);
    if (RC_size != 0)
    {
        DA      Á      Á
        blockAddr = 7; // data block 7
        status = MFRC522_Auth (PICC_AUTHENT1A, blockAddr, sectorKeyA [blockAddr /
        4],serNum); //
        if (status == MI_OK)
        {
            //
            status = MFRC522_Write (blockAddr, sectorNewKeyA [blockAddr / 4]);
            Serial.print ("set the new card password, and can modify the data of
            the Sector: ");
            Serial.print (blockAddr / 4, DEC);
            //
            blockAddr = blockAddr - 3;
            status = MFRC522_Write (blockAddr, writeDate);
            if (status == MI_OK)
            {
                Serial.println ("OK!");
            }
        }
    }
    //
    blockAddr = 7; // data block 7
    status = MFRC522_Auth (PICC_AUTHENT1A, blockAddr,
    sectorNewKeyA [blockAddr / 4], serNum); //
    if (status == MI_OK)
    {
        //
        blockAddr = blockAddr - 3;
        status = MFRC522_Read (blockAddr, str);
        if (status == MI_OK)
        {
            Serial.println ("Read from the card, the data is:");
            for (i = 0; i < 16; i++)
            {
                Serial.print (str [i]);
            }
            Serial.println ("");
        }
    }
}

```

```

Serial.println("");
MFRC522_Halt (); //
}
/*
 *      : Write_MFRC5200
 *      : MFRC522 of a register to write a byte of data
 *      : addr - register address; val - the value to be written
 */
void MFRC522::Write_MFRC5200 (uint8_t addr, uint8_t val) {
    SPI.beginTransaction(SPISettings(SPI_CLOCK_DIVIDER, SPI_MODE_0, SPI_MAX_TRANSFER_SIZE));
    digitalWrite(SS, LOW);
    SPI.transfer(addr << 2);
    SPI.transfer(val);
    digitalWrite(SS, HIGH);
    SPI.endTransaction();
}

```