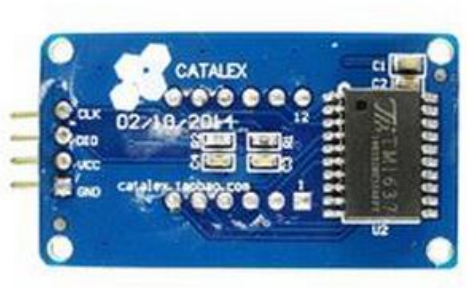
User guide for TM1637 4 digits display



Connection to an Arduino is like so:

VCC to Arduino 5v

GND to Arduino GND

CLK to Arduino digital pin, your choice

DIO to Arduino digital pin, your choice

This document is based on the library available at http://playground.arduino.cc/Main/TM1637.

The library defines a class object TM1637Display with public methods for controlling the device through two digital IO pins.

First you have to create a new object of type TM1637Display like so:

```
#include <TM1637Display.h>
#define CLK 9 //can be any digital pin
#define DIO 8 //can be any digital pin

TM1637Display display(CLK, DIO);
```

Available functions

display.setBrightness

You can set the brightness with the following command setting the parameter from 0 for lowest brightness to 7 for the highest.

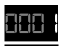
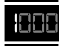
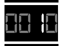
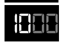
```
setBrightness(7);
```

display.showNumberDec

Next function that is simple to use is **showNumberDec**. The parameters are:

- Number of type integer. Values up to 9999.
- True/false. Display leading zeroes for values up to 999. Default value is false;
- Correspond to the number of digits to be displayed.
- Position of the least significant digit (0 – leftmost, 3 – rightmost).

Examples:

display.showNumberDec(1,false) 
display.showNumberDec(1,false,1,0) 
display.showNumberDec(1,false,1,2) 
display.showNumberDec(10,false,2,0) 

You can write a for example a for loop to display numbers from 1 to 9999.

```
void loop() {
  display.showNumberDec(i++)
  if (I > 9999)
    i=0;
}
```

To turn on the center colon , execute the following code.

```
uint8_t segto;
int value = 1244;
segto = 0x80 | display.encodeDigit((value / 100)%10);
display.setSegments(&segto, 1, 1);
```

display.setSegments

Each of the four digits has 7 segments that can be addressed each individually. Each segment is represented by a letter as shown below which is addressed by one bit.

```
//      A
//     ---
//  F |   | B
//     -G-
//  E |   | C
//     ---
//      D
```

The table below shows mapping between from numbers to segments.

```
    XGFEDCBA
  0b00111111,     0
  0b00000110,     1
  0b01011011,     2
  0b01001111,     3
  0b01100110,     4
  0b01101101,     5
  0b01111101,     6
  0b00000111,     7
  0b01111111,     8
  0b01101111,     9
  0b01110111,     A
  0b01111100,     B
  0b00111001,     C
  0b01000111,     D
  0b01111001,     E
  0b01110001      F
```

To turn off all segments, use this command.

```
Uint8_t data[] = {0x0, 0x0, 0x0, 0x0};
display.setSegments(data);
```

To turn on all segments, use this command.

```
Uint8_t data[] = {0xff, 0xff, 0xff, 0xff};
display.setSegments(data);
```

display.encodeDigit

This fonction returns the 7 bits representation for the digit between 1 and 15.

Uint8_t data[] = {0x0, 0x0, 0x0, 0x0};
data[0]= display.encodeDigit(15);
display.setSegments(data);

Result:

Note: Release version 1.0.0 has bug for the representation of letter D in the TM1637Display.cpp file. You need to change the code for this:

```
const uint8_t digitToSegment[] = {
 // XGFEDCBA
  0b00111111,    // 0
  0b00000110,    // 1
  0b01011011,    // 2
  0b01001111,    // 3
  0b01100110,    // 4
  0b01101101,    // 5
  0b01111101,    // 6
  0b00000111,    // 7
  0b01111111,    // 8
  0b01101111,    // 9
  0b01110111,    // A
  0b01111100,    // B
  0b00111001,    // C
  0b01011110,    // D
  0b01111001,    // E
  0b01110001     // F
  };
```