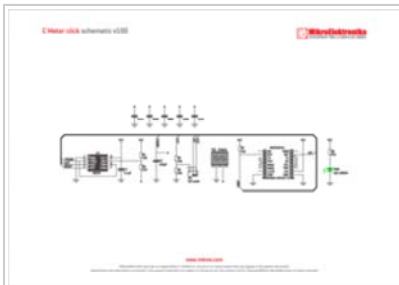


## C Meter click

From MikroElektronika Documentation

**C Meter click** has circuitry for measuring the value of capacitors. The design is based on a NE-555 timer/square-wave generator. The chip is implemented in an astable multivibrator circuit with fixed resistors and adjustable capacitors.

### Features and usage notes



Schematic also available in PDF ([http://cdn-docs.mikroe.com/images/d/dd/C\\_Meter\\_click\\_schem](http://cdn-docs.mikroe.com/images/d/dd/C_Meter_click_schem))

C Meter click has four different-sized plates for placing SMD capacitors (covering standard SMD packaging dimensions) as well as a socket for inserting a thru hole resistor. Each plate is labeled with package size in standard imperial measurements:

1210 — 3.2mm x 2.5mm

1206 — 3.2 mm x 1.6 mm

0805 — 2.0 mm x x1.25 mm

0603 — 1.6 mm x 0.8 mm

The board outputs a square wave frequency through the INT pin. Depending on the capacitor placed on the board, the value of this frequency changes. The value of a capacitor can be inferred using a simple algorithm (shown in the Libstock code example).

Onboard screw terminals are placed to allow the click to be used with oscillator probes.

C Meter click is designed to use a 5V power supply, but can work with either 3.3V or 5V logic levels.

C Meter click



C Meter click

<b>IC/Module</b>	NE-555 precision timer ( <a href="http://www.ti.com/lit/ds/symlink/ne555.pdf">http://www.ti.com/lit/ds/symlink/ne555.pdf</a> )
<b>Interface</b>	INT, RST
<b>Power supply</b>	5V (possible to use with 3.3V I/O)
<b>Website</b>	<a href="http://www.mikroe.com/click/c-meter">www.mikroe.com/click/c-meter</a> ( <a href="http://www.mikroe.com/click/c-meter">http://www.mikroe.com/click/c-meter</a> )

### Programming

This code snippet uses a preset calibration value and finds the capacitance of the capacitor for displaying on the TFT.

```

1 #include <stdint.h>
2 #include <stdbool.h>
3 #include "resources.h"
4
5 // C_Meter Click
6 sbit C_METER_RST at GPIOC_ODR.B2;
7 sbit C_METER_INT at GPIOD_ODR.B10;
8 // TFT module connections
9 unsigned int TFT_DataPort at GPIOE_ODR;
10 sbit TFT_RST at GPIOE_ODR.B8;
11 sbit TFT_RS at GPIOE_ODR.B12;
12 sbit TFT_CS at GPIOE_ODR.B15;
13 sbit TFT_RD at GPIOE_ODR.B10;
14 sbit TFT_WR at GPIOE_ODR.B11;
15 sbit TFT_BLED at GPIOE_ODR.B9;
16
17
18
19 void system_setup( void );
20 void setup_interrupt( void );
21 void InitTimer2( void );
22 void display_init( void );
23
24 uint32_t int_count = 0;
25 uint32_t timer_count_end = 0;
26 bool sec_flag = false;
27
28 void main()
29 {
30     //Local Declarations
31     char uart_text[ 40 ];
32     float c_cal = 0.0;
33     float t_meas = 0.0;
34     float c_meas = 0.0;
35     float ffinal = 0.0;
36     float t_cal = 0.0;
37
38     system_setup();
39     setup_interrupt();
40     InitTimer2();
41     display_init();
42
43     t_cal = 1.0 / 23350.0;

```

```

44  c_cal = t_cal / 77616.0;
45
46  int_count = 0;
47
48  while(1)
49  {
50
51      if( sec_flag )
52      {
53          //Measurement
54          t_meas = 1.0 / timer_count_end;
55          c_meas = t_meas / 77616.0;
56          final = fabs( c_cal - c_meas );
57          //Print out
58          FloatToStr( final, final_text );
59          TFT_Rectangle( 100, 100, 200, 120 );
60          TFT_Write_Text( final_text, 100, 100 );
61          TFT_Write_Text( " Farad", 190, 100 );
62          //Reset flags
63          sec_flag = false;
64          int_count = 0;
65          timer_count_end = 0;
66      }
67  }
68 }
69
70 void display_init( void )
71 {
72     TFT_Init_ILI9341_8bit( 320, 240 );
73     TFT_BLED = 1;
74     TFT_Set_Pen( CL_WHITE, 1 );
75     TFT_Set_Brush( 1, CL_WHITE, 0, 0, 0, 0 );
76     TFT_Set_Font( TFT_defaultFont, CL_BLACK, FO_HORIZONTAL );
77     TFT_Fill_Screen( CL_WHITE );
78     TFT_Set_Pen( CL_BLACK, 1 );
79     TFT_Line( 20, 46, 300, 46 );
80     TFT_Line( 20, 70, 300, 70 );
81     TFT_Line( 20, 220, 300, 220 );
82     TFT_Set_Pen( CL_WHITE, 1 );
83     TFT_Set_Font( &HandelGothic_BT21x22_Regular, CL_RED, FO_HORIZONTAL );
84     TFT_Write_Text( "C Meter click", 105, 14 );
85     TFT_Set_Font( &Tahoma15x16_Bold, CL_BLUE, FO_HORIZONTAL );
86     TFT_Write_Text( "C Meter", 135, 50 );
87     TFT_Set_Font( &Verdana12x13_Regular, CL_BLACK, FO_HORIZONTAL );
88     TFT_Write_Text( "EasyMx PRO v7 for STM32", 19, 223 );
89     TFT_Set_Font( &Verdana12x13_Regular, CL_RED, FO_HORIZONTAL );
90     TFT_Write_Text( "www.mikroe.com", 200, 223 );
91     TFT_Set_Font( &Tahoma15x16_Bold, CL_BLACK, FO_HORIZONTAL );
92 }
93
94 void system_setup( void )
95 {
96     //GPIO
97     GPIO_Digital_Output( &GPIOC_BASE, _GPIO_PINMASK_2 ); //RST
98     GPIO_Digital_Input( &GPIOD_BASE, _GPIO_PINMASK_10 ); //INT
99
100    //RST Toggle
101    C_METER_RST = 0;
102    Delay_ms(50);
103    C_METER_RST = 1;
104
105    //UART
106    UART1_Init( 9600 );
107    UART1_Write_Text( "UART Initialized\r\n" );
108 }
109
110 void setup_interrupt( void )
111 {
112     GPIO_Digital_Output(&GPIOE_BASE, _GPIO_PINMASK_HIGH); // Enable digital output on PORTD
113     GPIOE_ODR = 0xAAAA;
114     GPIO_Digital_Input(&GPIOD_BASE, _GPIO_PINMASK_10);
115
116     RCC_APB2ENR.AFIOEN = 1; // Enable clock for alternate pin functions
117     AFIO_EXTICR3 = 0x0300; // PD10 as External interrupt
118     EXTI_FTSR = 0x00000400; // Set interrupt on Rising edge
119     EXTI_TMR |= 0x00000400; // Set mask
120     NVIC_IntEnable(IVT_INT_EXTI15_10); // Enable External interrupt
121     EnableInterrupts(); // Enables the processor interrupt.
122 }
123
124
125 void ExtInt() iv IVT_INT_EXTI15_10 ics ICS_AUTO
126 {
127     EXTI_PR.B10 = 1; // clear flag
128     int_count++;
129 }
130
131 void InitTimer2() //1 second
132 {
133     RCC_APB1ENR.TIM2EN = 1;
134     TIM2_CR1.CEN = 0;
135     TIM2_PSC = 1124;
136     TIM2_ARR = 63999;
137     NVIC_IntEnable(IVT_INT_TIM2);
138     TIM2_DIER.UIE = 1;
139     TIM2_CR1.CEN = 1;
140 }
141
142 void Timer2_interrupt() iv IVT_INT_TIM2
143 {
144     TIM2_SR.UIF = 0;
145     timer_count_end = int_count;
146     int_count = 0;
147     sec_flag = true;
148 }

```

Code examples that demonstrate the usage of C Meter click with MikroElektronika hardware, written for mikroC for ARM is available on Libstock (<http://libstock.mikroe.com/projects/view/1879/c-meter-click>).

## Resources

- C Meter click example on Libstock (<http://libstock.mikroe.com/projects/view/1879/c-meter-click>)
- NE-555 vendor's data sheet (<http://www.ti.com/lit/ds/symlink/ne555.pdf>)
- mikroBUS standard specifications (<http://download.mikroe.com/documents/standards/mikrobus/mikrobus-standard-specification-v200.pdf>)

Retrieved from "[http://docs.mikroe.com/index.php?title=C\\_Meter\\_click&oldid=601](http://docs.mikroe.com/index.php?title=C_Meter_click&oldid=601)"

- 
- This page was last modified on 1 August 2016, at 17:04.
  - Content is available under Creative Commons Attribution unless otherwise noted.