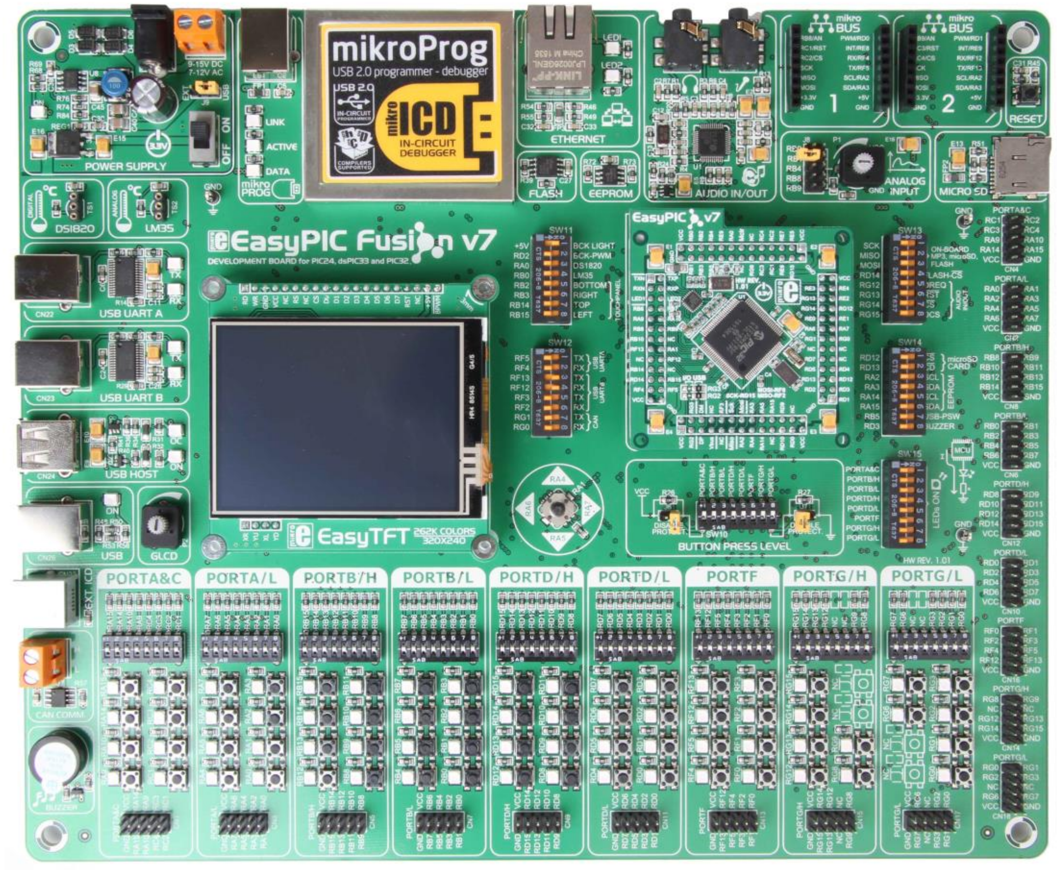
IoT Development Tools for PIC32



When Bluetooth and Wi-Fi Just Won't Do

February 2, 2018

FRED EADY

Presented by:

DesignNews

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

IoT Development Tools for PIC32

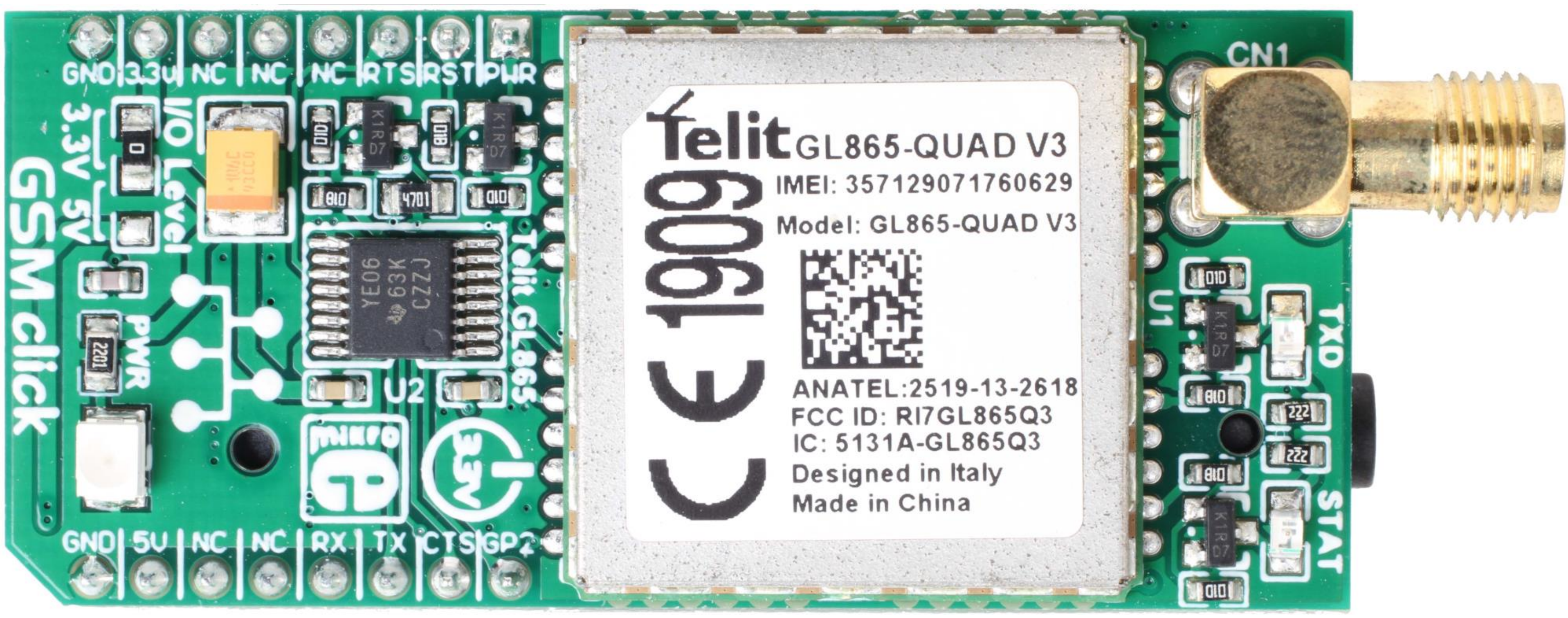
AGENDA

- MikroElektronika Hardware
- Sending an SMS Message
- Adios Amigos

IoT Development Tools for PIC32

MikroElektronika Hardware – click Part Number

ANTENNA GSM SMA RA BGSM CONN
MIKROE-275
1471-1168-ND

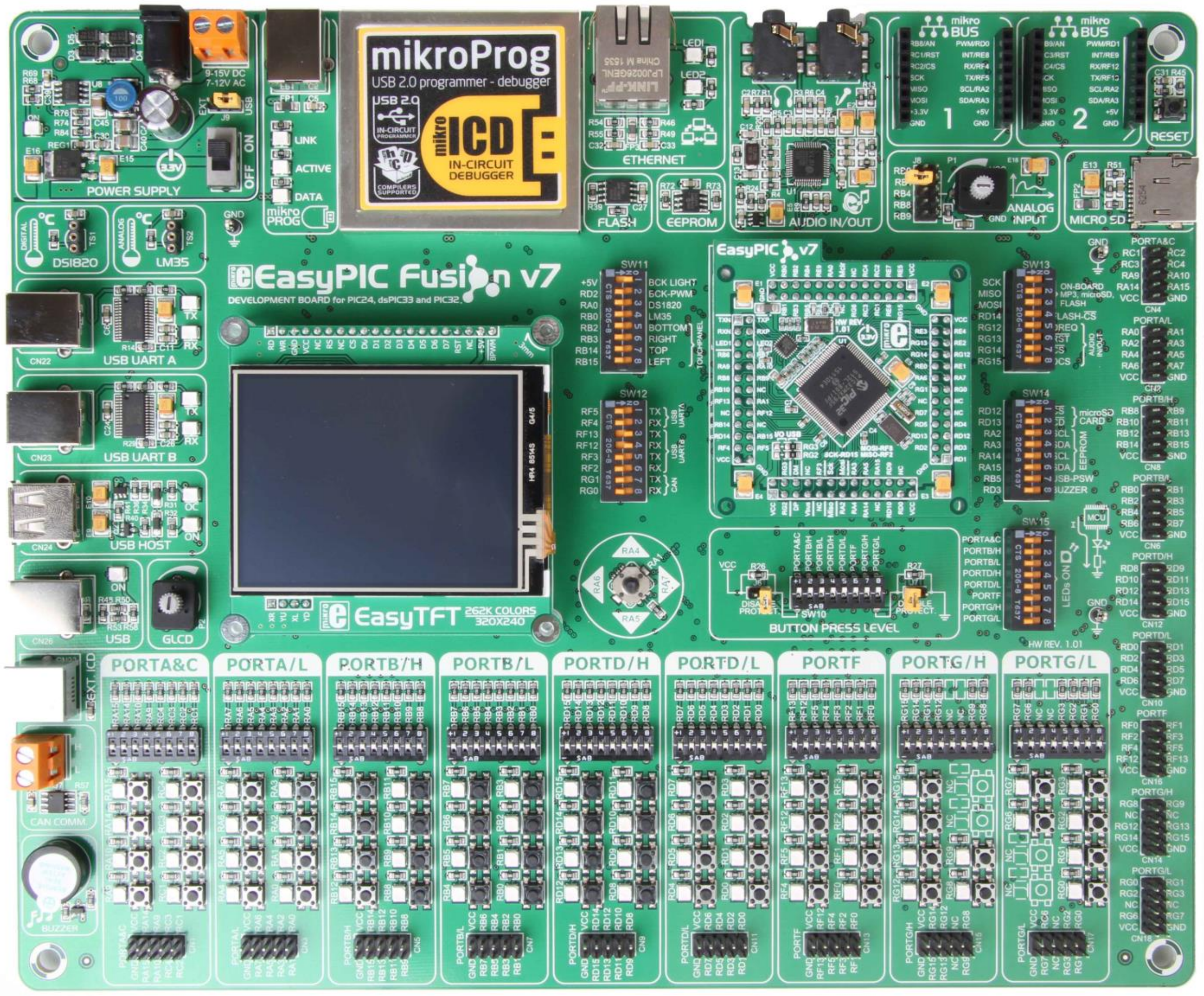


RF TXRX MODULE CELLULAR
MIKROE-1298
1471-1065-ND

Presented by:

IoT Development Tools for PIC32

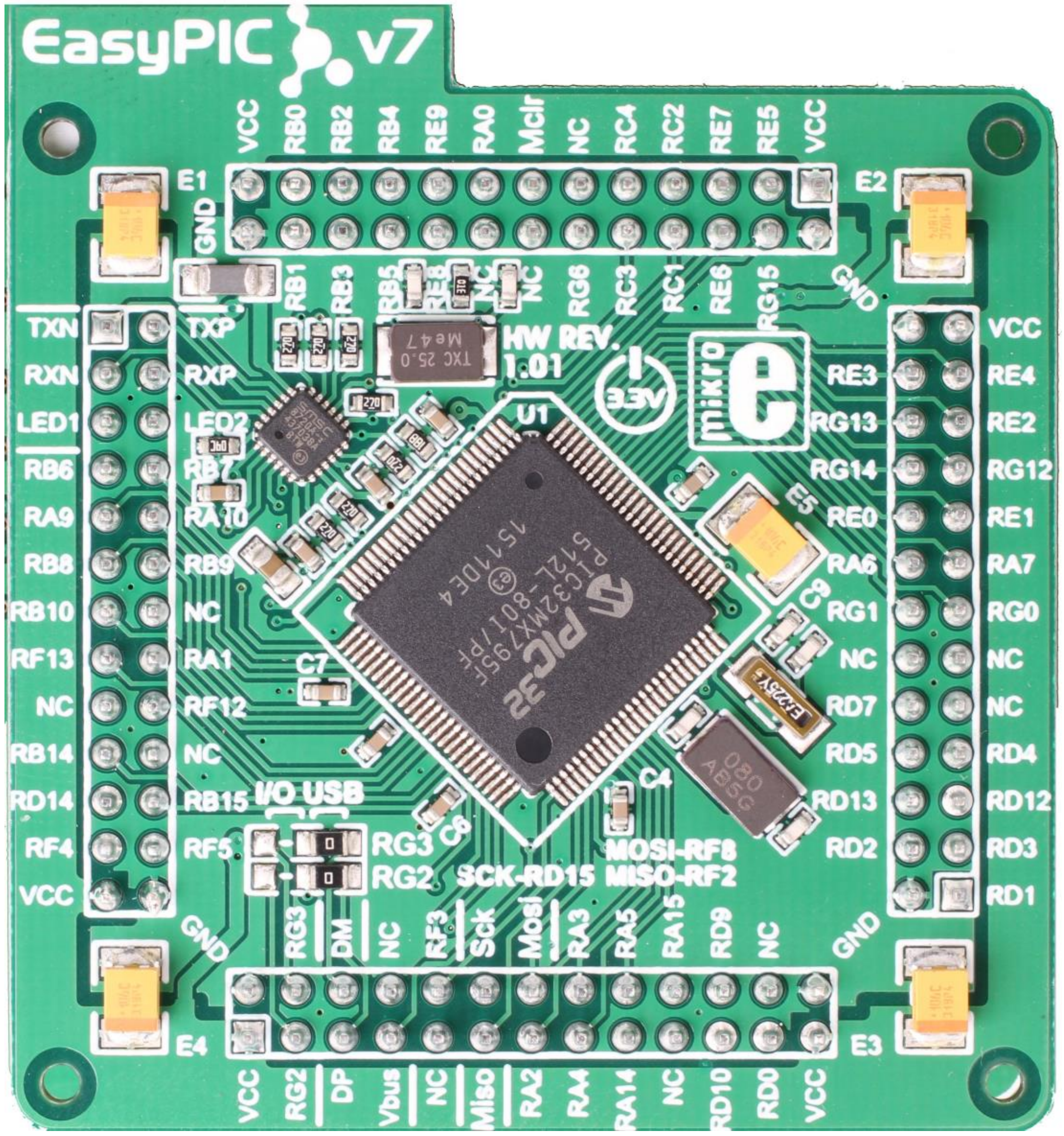
MikroElektronika Hardware – EasyPIC Fusion v7



Presented by:

IoT Development Tools for PIC32

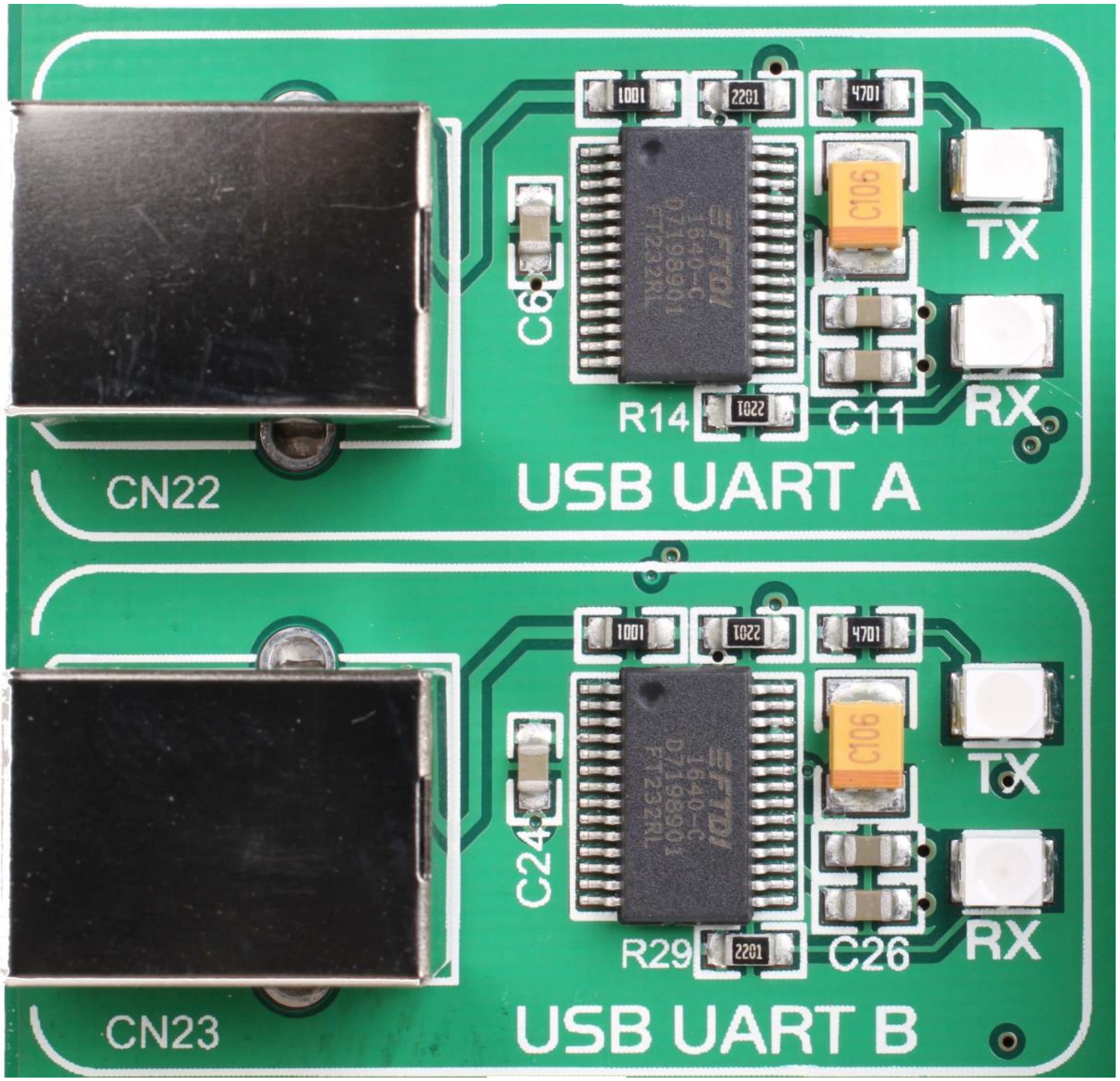
MikroElektronika Hardware – EasyPIC Fusion v7


EasyPIC v7
VCC RB0 RB2 RB4 RE9 RA0 Mclr NC RC4 RC2 RE7 RE5 VCC
E1 E2 GND
RB1 RB3 RB5 RE8 NC NC RG6 RC3 RC1 RE6 RG15
TXN RXN LED1 RB6 RA9 RB8 RB10 RF13 NC RB14 RD14 RF4 VCC
TXP RXP LED2 RB7 RA10 RB9 NC RA1 RF12 NC RB15 RF5
HW REV. 1.01
U1
3.3V
VCC RE4 RE2 RG12 RE1 RA7 RG0 NC NC RD4 RD12 RD3 RD1
RE3 RG13 RG14 RE0 RA6 RG1 NC RD7 RD5 RD13 RD2
E5 C8 C7 C6 C4
PIC32 PIC32MX795F 512L-80I/PF
I/O USB
RG3 RG2 SCK-RD15 MOSI-RF8 MISO-RF2
RG3 DM NC RF3 Sck Mosi RA3 RA5 RA15 RD9 NC
E4 E3
VCC RG2 DP Vbus NC Miso RA2 RA4 RA14 NC RD10 RD0 VCC

Presented by:

IoT Development Tools for PIC32

MikroElektronika Hardware – EasyPIC Fusion v7



Presented by:

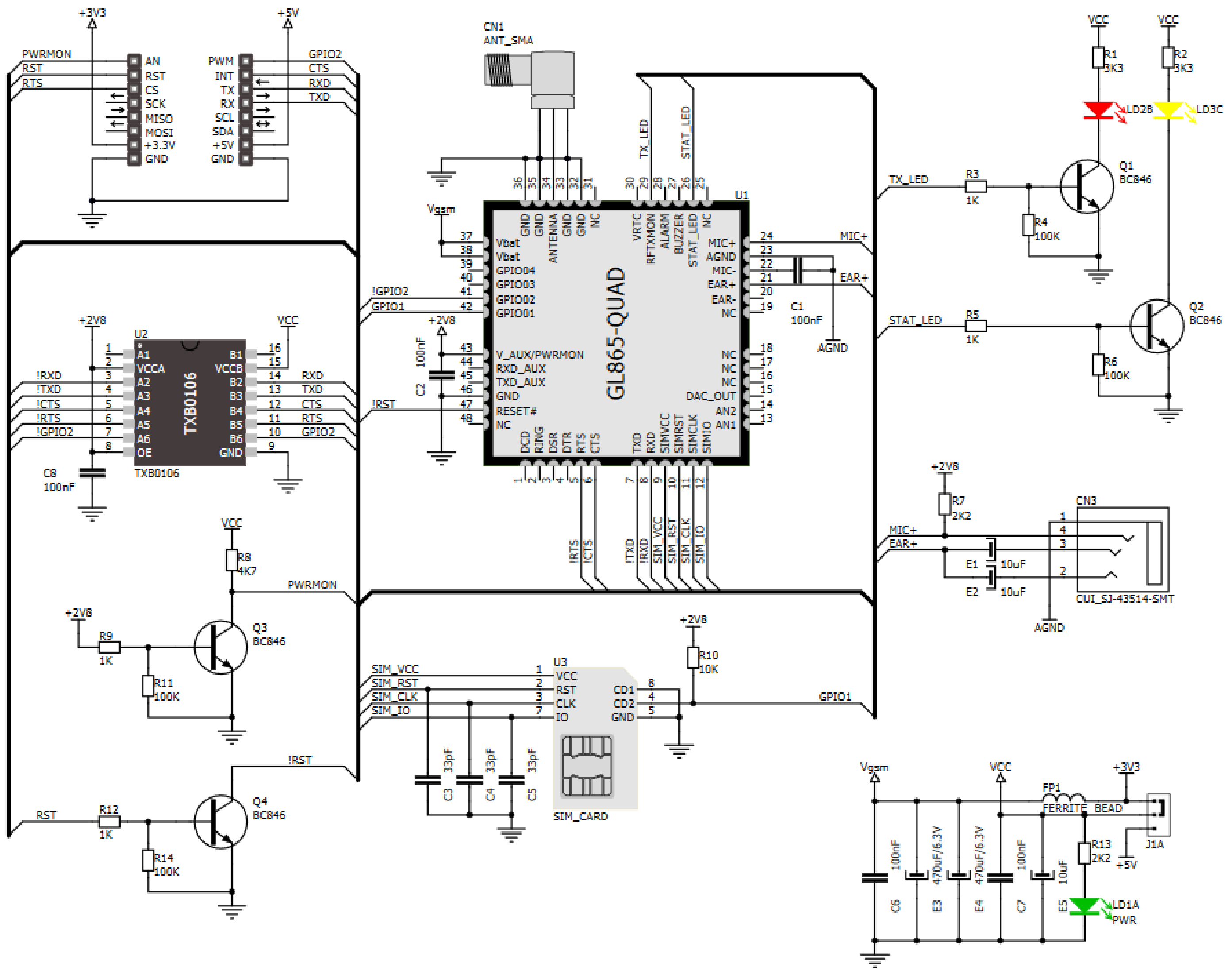
IoT Development Tools for PIC32

MikroElektronika Hardware – EasyPIC Fusion v7

Presented by:

IoT Development Tools for PIC32

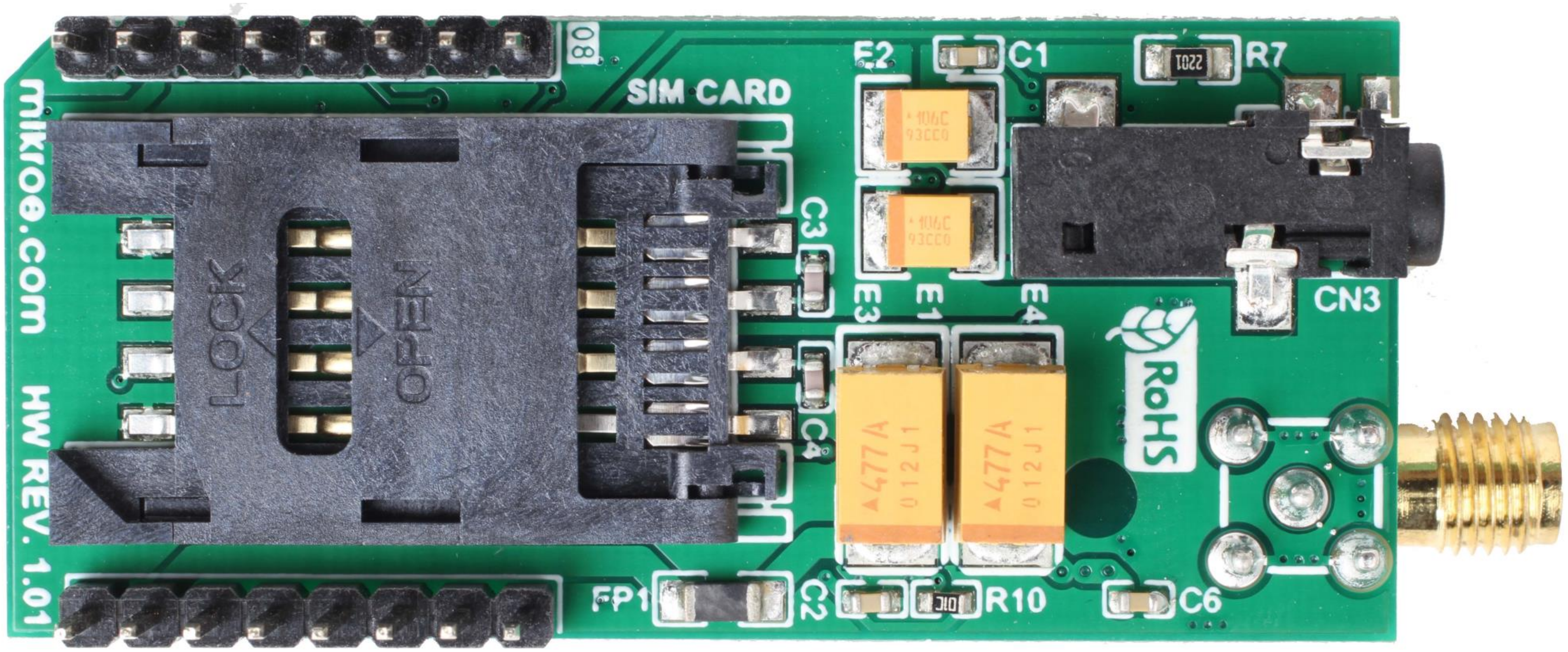
MikroElektronika Hardware – GSM click



Presented by:

IoT Development Tools for PIC32

MikroElektronika Hardware – GSM click



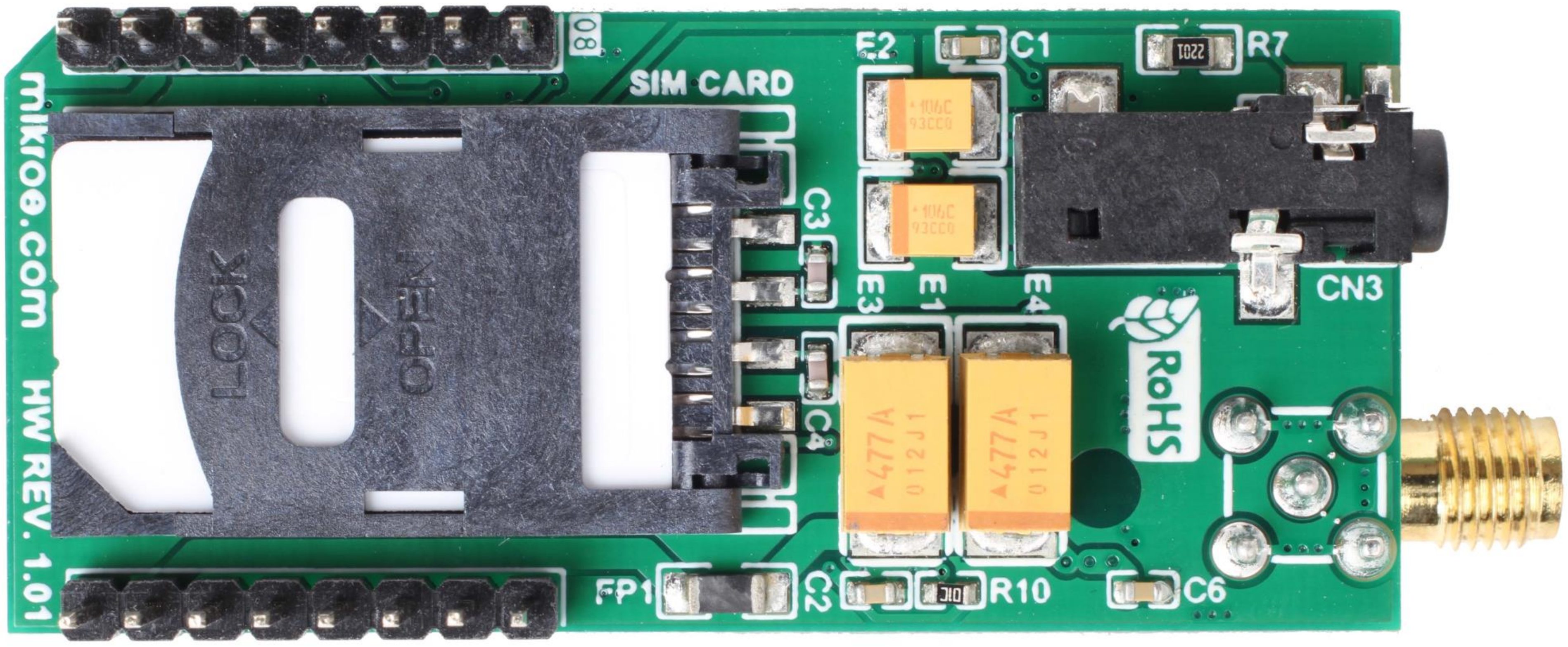
Presented by:

IoT Development Tools for PIC32

MikroElektronika Hardware – GSM click

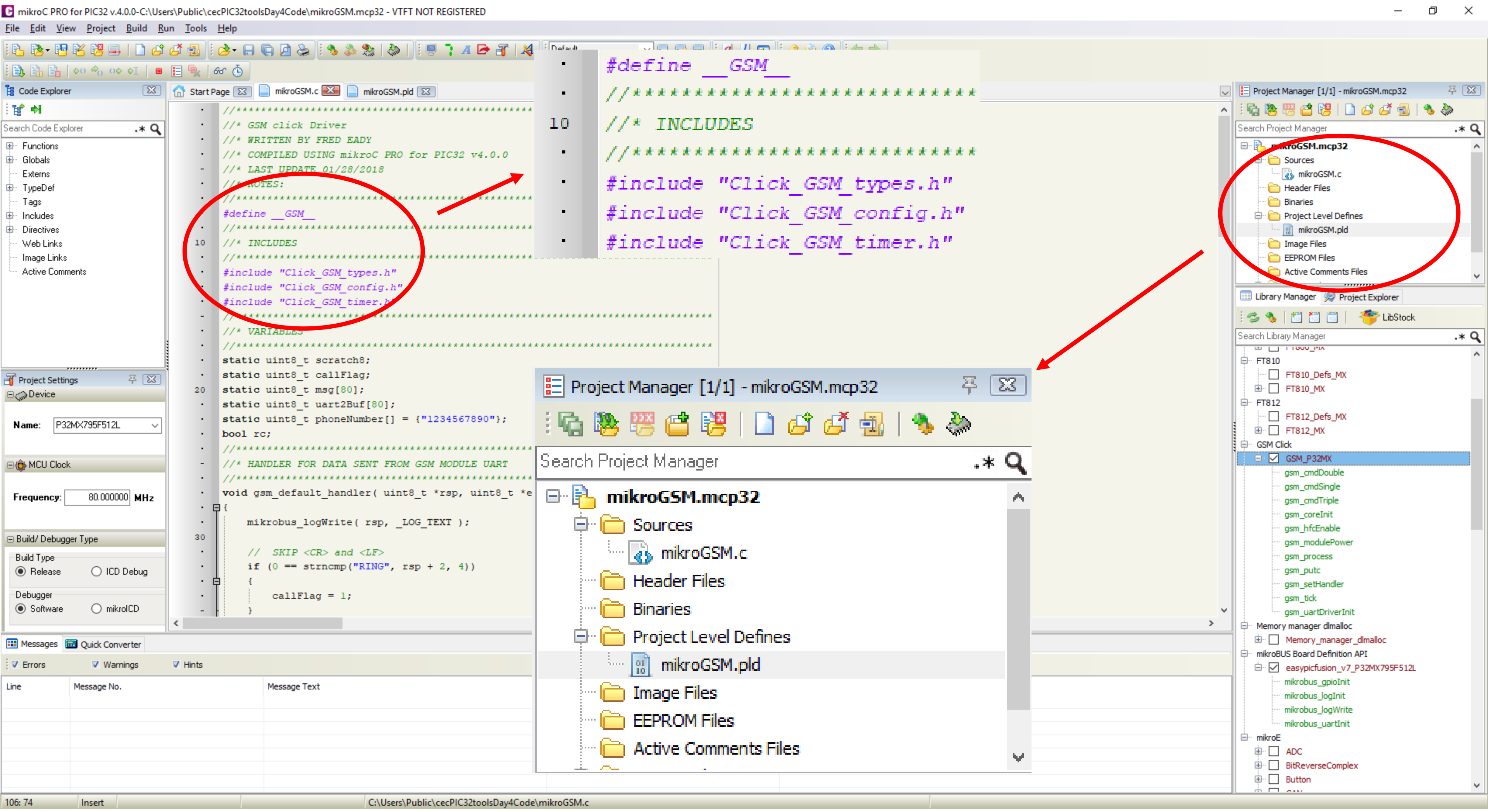
IoT Development Tools for PIC32

MikroElektronika Hardware – GSM click



IoT Development Tools for PIC32

Sending an SMS Message – mikroC PRO for 32



mikroC PRO for PIC32 v.4.0.0-C:\Users\Public\cecPIC32toolsDay4Code\mikroGSM.mcp32 - VTFT NOT REGISTERED

```
//*********************************************
//* GSM click Driver
//* WRITTEN BY FRED EADY
//* COMPILED USING mikroC PRO for PIC32 v4.0.0
//* LAST UPDATE 01/28/2018
//* NOTES:
//*********************************************
#define __GSM__
//*********************************************
//* INCLUDES
//*********************************************
#include "Click_GSM_types.h"
#include "Click_GSM_config.h"
#include "Click_GSM_timer.h"
//*********************************************
//* VARIABLES
//*********************************************
static uint8_t scratch8;
static uint8_t callFlag;
static uint8_t msg[80];
static uint8_t uart2Buf[80];
static uint8_t phoneNumber[] = {"1234567890"};
bool rc;
//*********************************************
//* HANDLER FOR DATA SENT FROM GSM MODULE UART
//*********************************************
void gsm_default_handler( uint8_t *rsp, uint8_t *e
{
    mikrobus_logWrite( rsp, _LOG_TEXT );

    //  SKIP <CR> and <LF>
    if (0 == strncmp("RING", rsp + 2, 4))
    {
        callFlag = 1;
    }
```

```
#define __GSM__
//*****************************
//* INCLUDES
//*****************************
#include "Click_GSM_types.h"
#include "Click_GSM_config.h"
#include "Click_GSM_timer.h"
```

Project Settings – Device – Name: P32MX795F512L

MCU Clock – Frequency: 80.000000 MHz

Build/ Debugger Type – Build Type: [✓ Release] ICD Debug – Debugger: [✓ Software] mikroICD

Project Manager [1/1] - mikroGSM.mcp32

- mikroGSM.mcp32
  - Sources
    - mikroGSM.c
  - Header Files
  - Binaries
  - Project Level Defines
    - mikroGSM.pld
  - Image Files
  - EEPROM Files
  - Active Comments Files

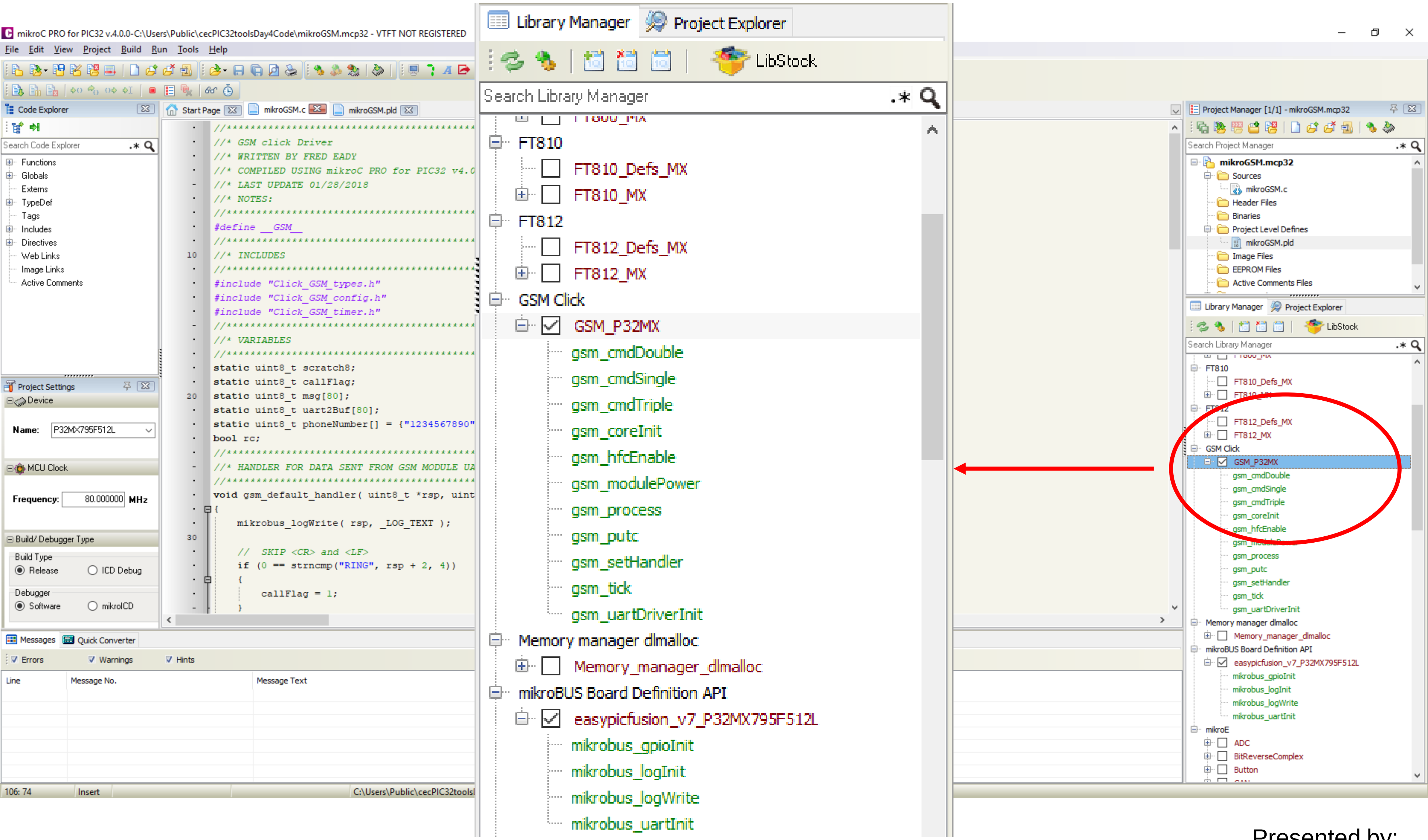
Library Manager – GSM Click – GSM_P32MX: gsm_cmdDouble, gsm_cmdSingle, gsm_cmdTriple, gsm_coreInit, gsm_hfcEnable, gsm_modulePower, gsm_process, gsm_putc, gsm_setHandler, gsm_tick, gsm_uartDriverInit; Memory manager dlmalloc – Memory_manager_dlmalloc; mikroBUS Board Definition API – easypicfusion_v7_P32MX795F512L: mikrobus_gpioInit, mikrobus_logInit, mikrobus_logWrite, mikrobus_uartInit

106: 74 Insert C:\Users\Public\cecPIC32toolsDay4Code\mikroGSM.c

Presented by:

IoT Development Tools for PIC32

Sending an SMS Message – mikroC PRO for 32



mikroC PRO for PIC32 v.4.0.0-C:\Users\Public\cecPIC32toolsDay4Code\mikroGSM.mcp32 - VTFT NOT REGISTERED

```
//*********************************************
//* GSM click Driver
//* WRITTEN BY FRED EADY
//* COMPILED USING mikroC PRO for PIC32 v4.0
//* LAST UPDATE 01/28/2018
//* NOTES:
//*********************************************
#define __GSM__
//*********************************************
//* INCLUDES
//*********************************************
#include "Click_GSM_types.h"
#include "Click_GSM_config.h"
#include "Click_GSM_timer.h"
//*********************************************
//* VARIABLES
//*********************************************
static uint8_t scratch8;
static uint8_t callFlag;
static uint8_t msg[80];
static uint8_t uart2Buf[80];
static uint8_t phoneNumber[] = {"1234567890"
bool rc;
//*********************************************
//* HANDLER FOR DATA SENT FROM GSM MODULE UA
//*********************************************
void gsm_default_handler( uint8_t *rsp, uint
{
    mikrobus_logWrite( rsp, _LOG_TEXT );

    //  SKIP <CR> and <LF>
    if (0 == strncmp("RING", rsp + 2, 4))
    {
        callFlag = 1;
    }
```

Project Settings – Device – Name: P32MX795F512L

MCU Clock – Frequency: 80.000000 MHz

Build/ Debugger Type – Build Type: [✓ Release] ICD Debug – Debugger: [✓ Software] mikroICD

Project Manager [1/1] - mikroGSM.mcp32: mikroGSM.mcp32 – Sources: mikroGSM.c; Header Files; Binaries; Project Level Defines: mikroGSM.pld; Image Files; EEPROM Files; Active Comments Files

Library Manager

- FT810
  - ☐ FT810_Defs_MX
  - ☐ FT810_MX
- FT812
  - ☐ FT812_Defs_MX
  - ☐ FT812_MX
- GSM Click
  - ☑ GSM_P32MX
    - gsm_cmdDouble
    - gsm_cmdSingle
    - gsm_cmdTriple
    - gsm_coreInit
    - gsm_hfcEnable
    - gsm_modulePower
    - gsm_process
    - gsm_putc
    - gsm_setHandler
    - gsm_tick
    - gsm_uartDriverInit
- Memory manager dlmalloc
  - ☐ Memory_manager_dlmalloc
- mikroBUS Board Definition API
  - ☑ easypicfusion_v7_P32MX795F512L
    - mikrobus_gpioInit
    - mikrobus_logInit
    - mikrobus_logWrite
    - mikrobus_uartInit

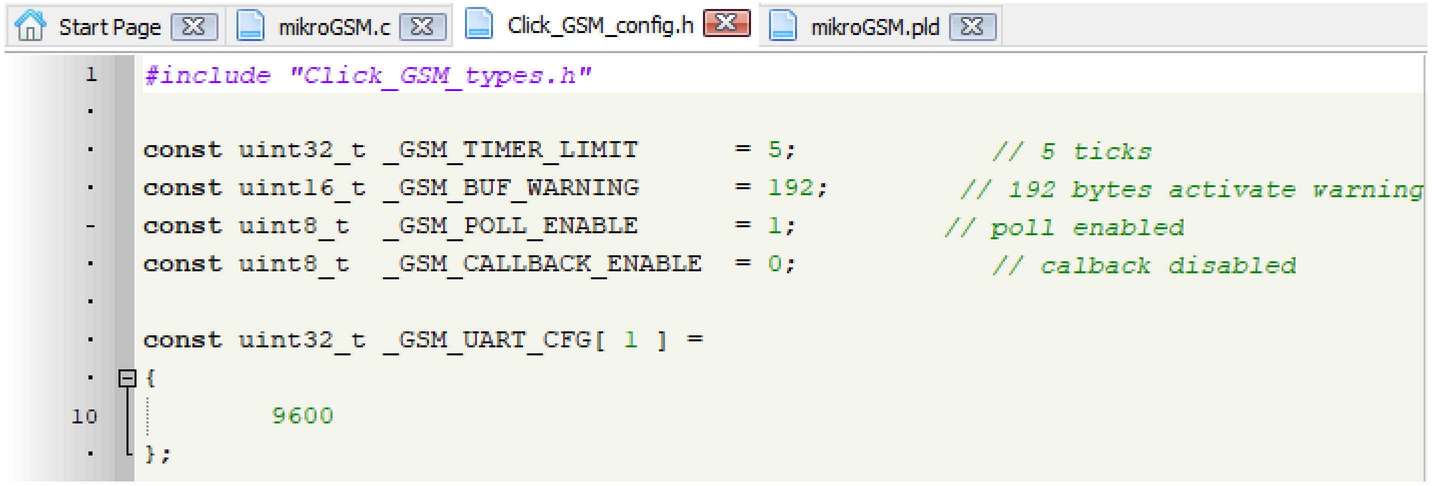
DesignNews

CEC CONTINUING EDUCATION CENTER

Presented by: Digi-Key ELECTRONICS

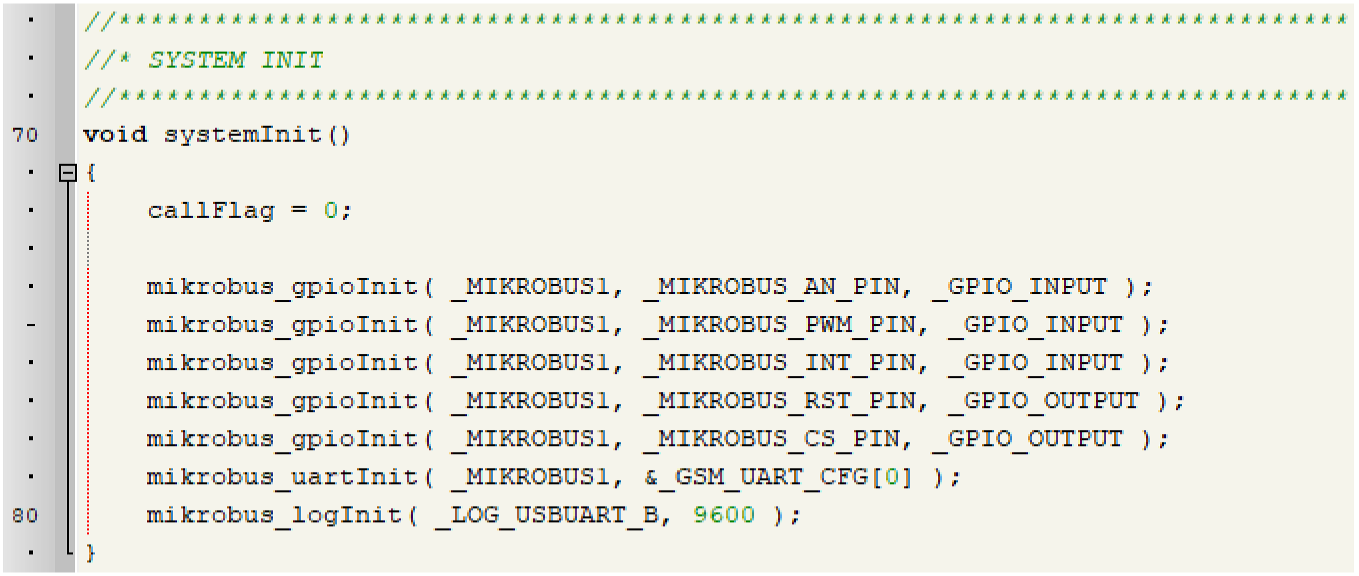
IoT Development Tools for PIC32

Sending an SMS Message – System Init Function



Start Page | mikroGSM.c | Click_GSM_config.h | mikroGSM.pld

```
#include "Click_GSM_types.h"

const uint32_t _GSM_TIMER_LIMIT      = 5;           // 5 ticks
const uint16_t _GSM_BUF_WARNING      = 192;       // 192 bytes activate warning
const uint8_t  _GSM_POLL_ENABLE      = 1;        // poll enabled
const uint8_t  _GSM_CALLBACK_ENABLE  = 0;           // calback disabled

const uint32_t _GSM_UART_CFG[ 1 ] =
{
        9600
};
```



```
//*****************************************************************************
//* SYSTEM INIT
//*****************************************************************************
void systemInit()
{
    callFlag = 0;

    mikrobus_gpioInit( _MIKROBUS1, _MIKROBUS_AN_PIN, _GPIO_INPUT );
    mikrobus_gpioInit( _MIKROBUS1, _MIKROBUS_PWM_PIN, _GPIO_INPUT );
    mikrobus_gpioInit( _MIKROBUS1, _MIKROBUS_INT_PIN, _GPIO_INPUT );
    mikrobus_gpioInit( _MIKROBUS1, _MIKROBUS_RST_PIN, _GPIO_OUTPUT );
    mikrobus_gpioInit( _MIKROBUS1, _MIKROBUS_CS_PIN, _GPIO_OUTPUT );
    mikrobus_uartInit( _MIKROBUS1, &_GSM_UART_CFG[0] );
    mikrobus_logInit( _LOG_USBUART_B, 9600 );
}
```

DesignNews

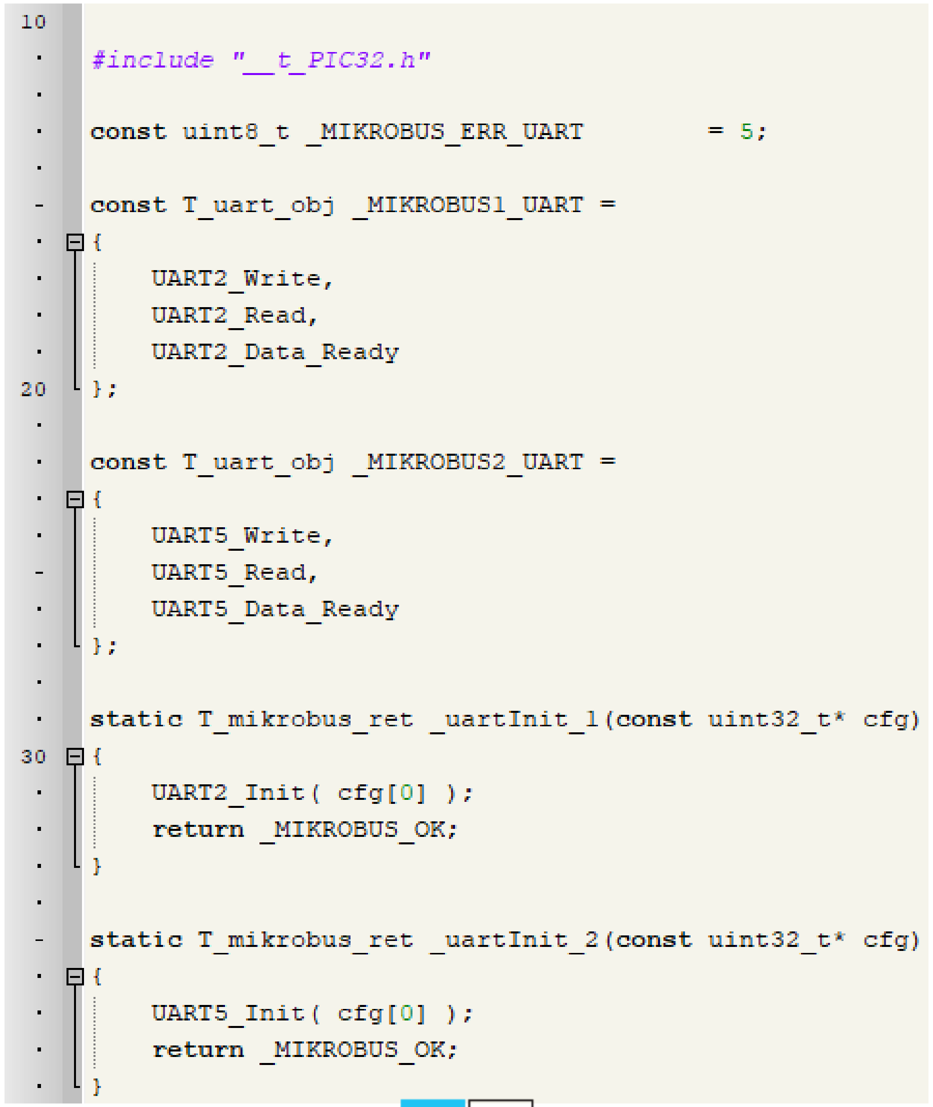
mikroSDK

CEC CONTINUING EDUCATION CENTER

Presented by: Digi-Key ELECTRONICS

IoT Development Tools for PIC32

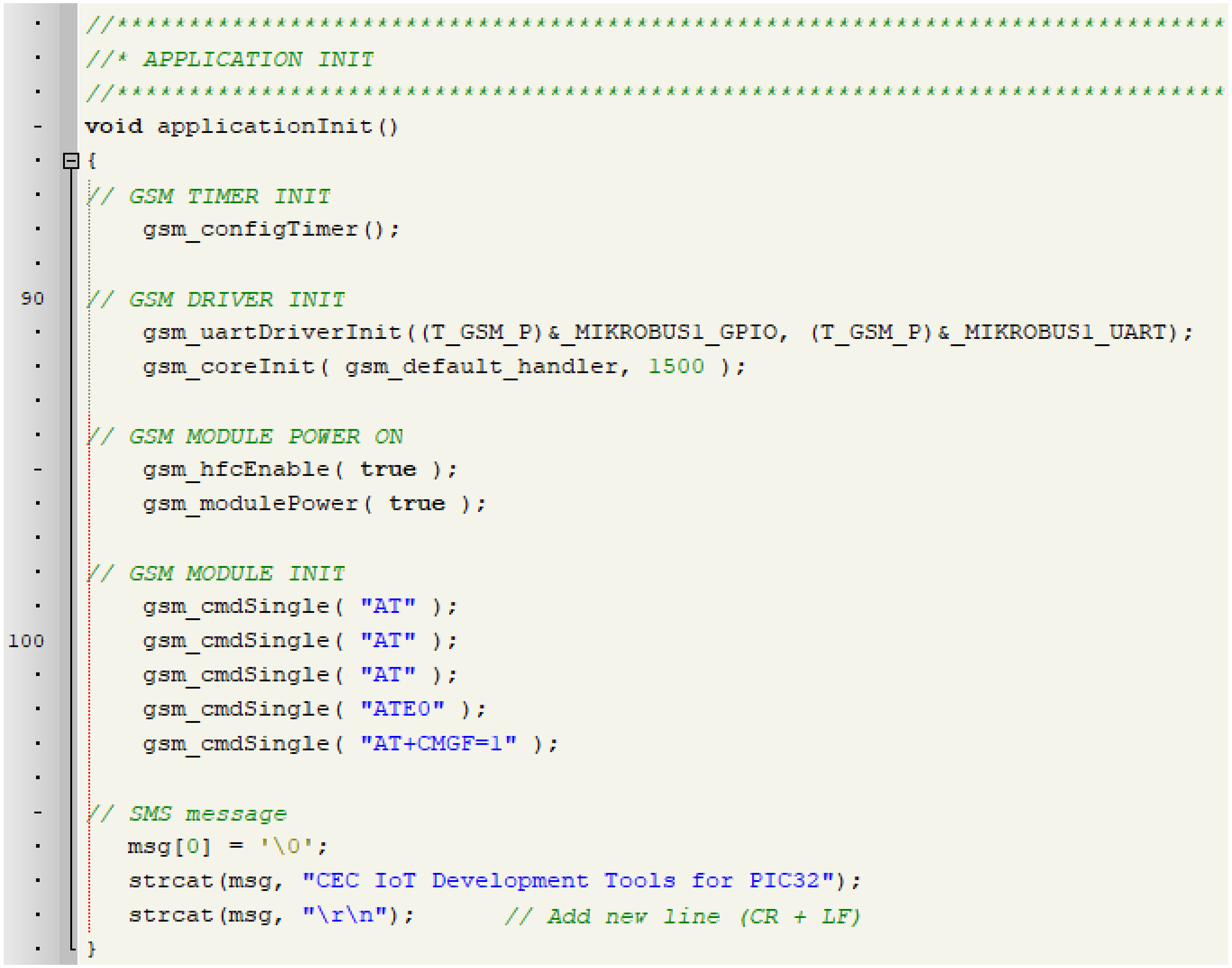
Sending an SMS Message – Which UART?



```
#include "__t_PIC32.h"

const uint8_t _MIKROBUS_ERR_UART        = 5;

const T_uart_obj _MIKROBUS1_UART =
{
    UART2_Write,
    UART2_Read,
    UART2_Data_Ready
};

const T_uart_obj _MIKROBUS2_UART =
{
    UART5_Write,
    UART5_Read,
    UART5_Data_Ready
};

static T_mikrobus_ret _uartInit_1(const uint32_t* cfg)
{
    UART2_Init( cfg[0] );
    return _MIKROBUS_OK;
}

static T_mikrobus_ret _uartInit_2(const uint32_t* cfg)
{
    UART5_Init( cfg[0] );
    return _MIKROBUS_OK;
}
```

IoT Development Tools for PIC32

Sending an SMS Message – Application Init Function

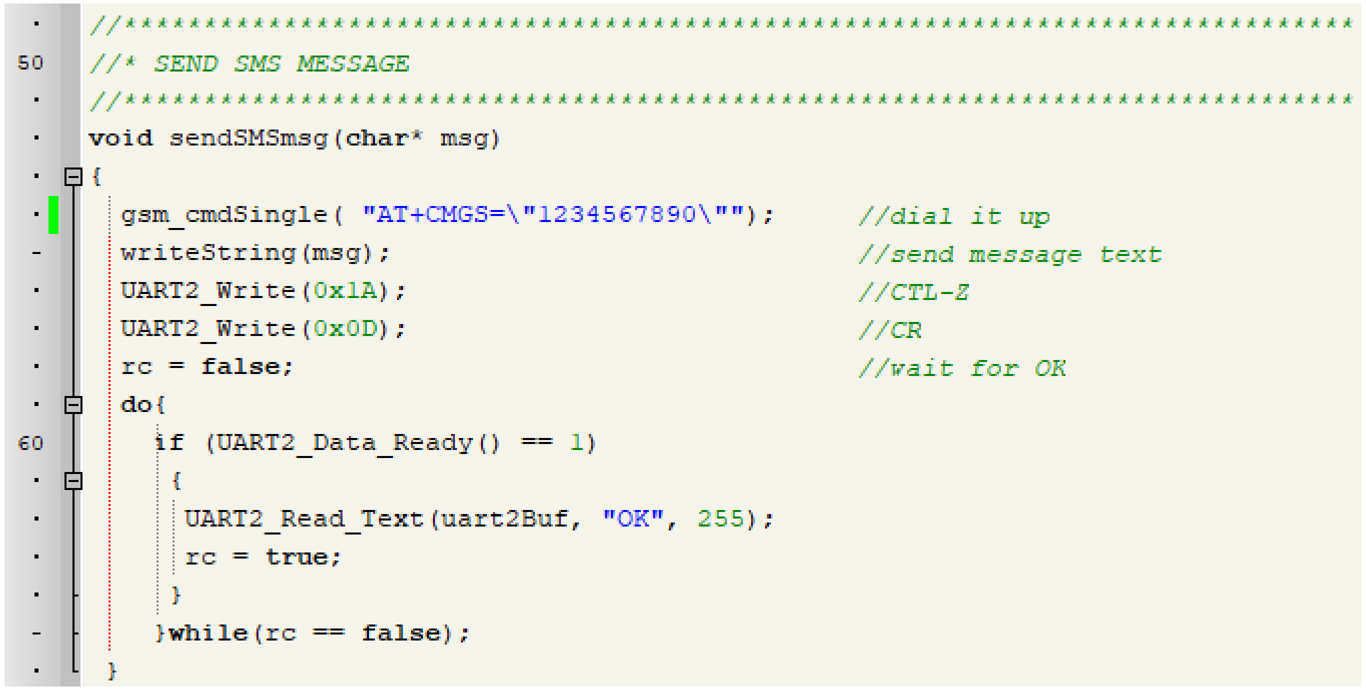


```
//***********************************************************************
//* APPLICATION INIT
//***********************************************************************
void applicationInit()
{
// GSM TIMER INIT
    gsm_configTimer();

// GSM DRIVER INIT
    gsm_uartDriverInit((T_GSM_P)&_MIKROBUS1_GPIO, (T_GSM_P)&_MIKROBUS1_UART);
    gsm_coreInit( gsm_default_handler, 1500 );

// GSM MODULE POWER ON
    gsm_hfcEnable( true );
    gsm_modulePower( true );

// GSM MODULE INIT
    gsm_cmdSingle( "AT" );
    gsm_cmdSingle( "AT" );
    gsm_cmdSingle( "AT" );
    gsm_cmdSingle( "ATE0" );
    gsm_cmdSingle( "AT+CMGF=1" );

// SMS message
   msg[0] = '\0';
   strcat(msg, "CEC IoT Development Tools for PIC32");
   strcat(msg, "\r\n");       // Add new line (CR + LF)
}
```

Presented by:

IoT Development Tools for PIC32

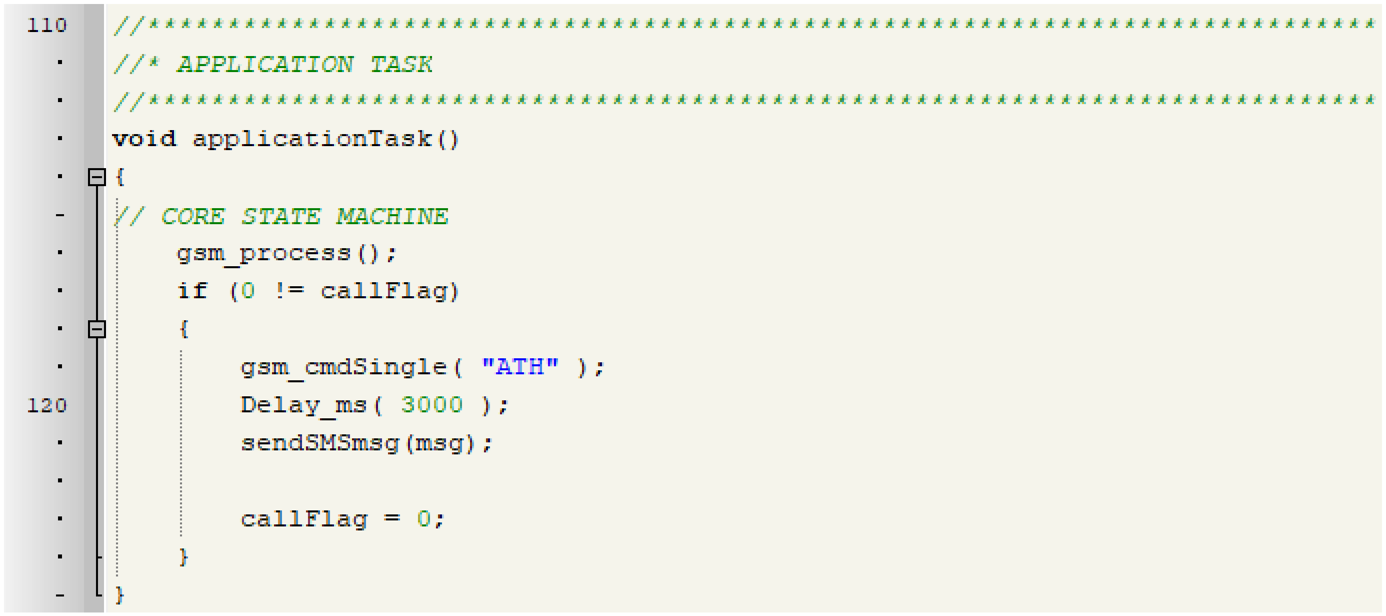
Sending an SMS Message – Send SMS Message Function



```
//**********************************************************************
//* SEND SMS MESSAGE
//**********************************************************************
void sendSMSmsg(char* msg)
{
   gsm_cmdSingle( "AT+CMGS=\"1234567890\"");      //dial it up
   writeString(msg);                              //send message text
   UART2_Write(0x1A);                             //CTL-Z
   UART2_Write(0x0D);                             //CR
   rc = false;                                    //wait for OK
   do{
     if (UART2_Data_Ready() == 1)
      {
       UART2_Read_Text(uart2Buf, "OK", 255);
       rc = true;
      }
     }while(rc == false);
 }
```

IoT Development Tools for PIC32

Sending an SMS Message – Send SMS Message Function

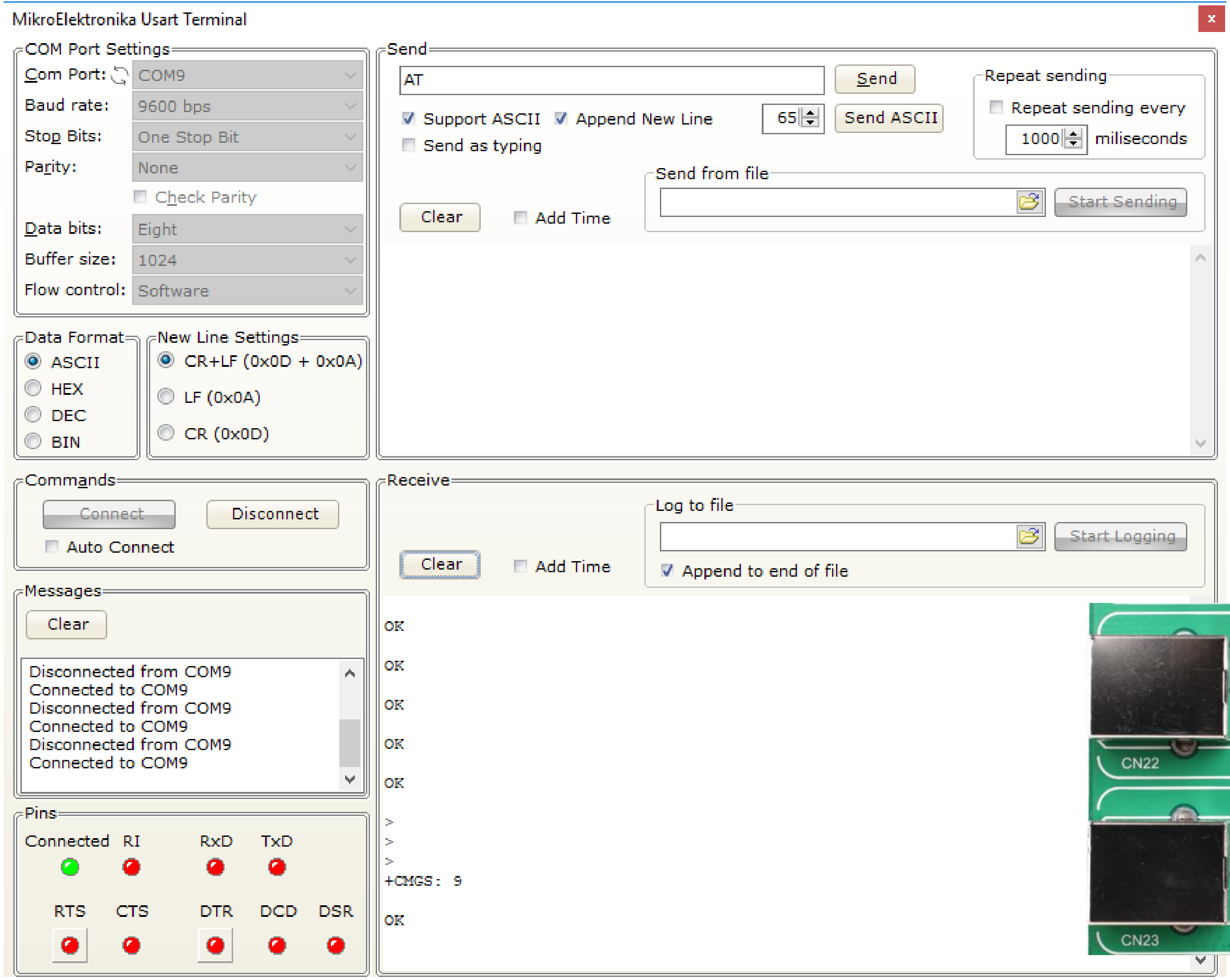


```
//*************************************************************************
//* APPLICATION TASK
//*************************************************************************
void applicationTask()
{
// CORE STATE MACHINE
    gsm_process();
    if (0 != callFlag)
    {
        gsm_cmdSingle( "ATH" );
        Delay_ms( 3000 );
        sendSMSmsg(msg);

        callFlag = 0;
    }
}
```

Presented by:

IoT Development Tools for PIC32

Sending an SMS Message – Debugging Tool



MikroElektronika Usart Terminal

COM Port Settings
- Com Port: COM9
- Baud rate: 9600 bps
- Stop Bits: One Stop Bit
- Parity: None
- Check Parity: ☐
- Data bits: Eight
- Buffer size: 1024
- Flow control: Software

Data Format: [✓ ASCII] HEX DEC BIN

New Line Settings: [✓ CR+LF (0x0D + 0x0A)] LF (0x0A) CR (0x0D)

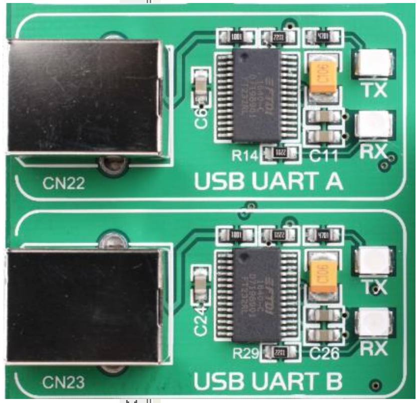
Send: AT
- ☑ Support ASCII ☑ Append New Line 65
- ☐ Send as typing
- ☐ Add Time
- Repeat sending: ☐ Repeat sending every 1000 miliseconds

Commands
- ☐ Auto Connect

Messages
```
Disconnected from COM9
Connected to COM9
Disconnected from COM9
Connected to COM9
Disconnected from COM9
Connected to COM9
```

Receive
- ☐ Add Time
- Log to file: ☑ Append to end of file

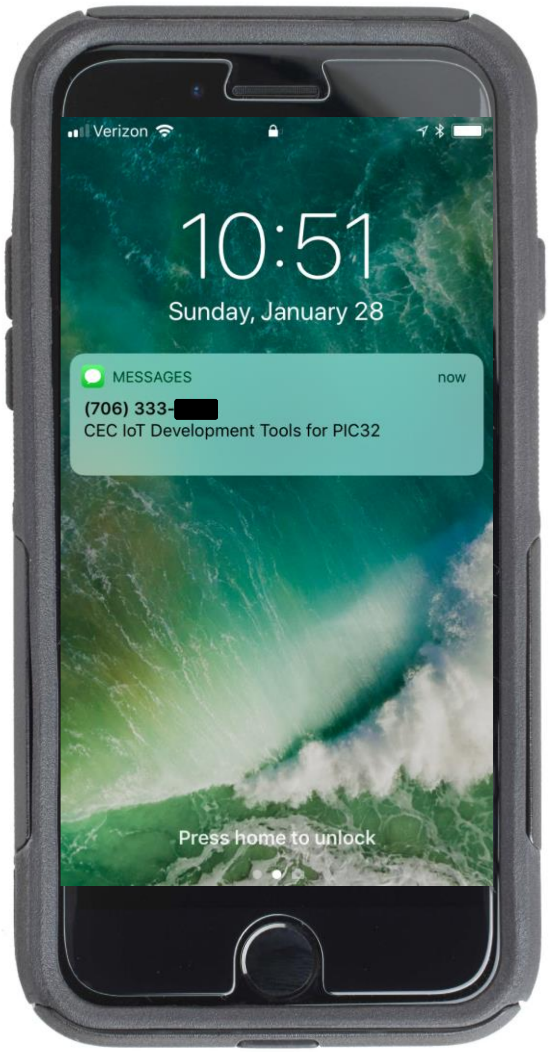
```
OK

OK

OK

OK

OK

>
>
>
+CMGS: 9

OK
```



Presented by:

IoT Development Tools for PIC32

Sending an SMS Message – Sent and Received!



(706) 333-[redacted]
CEC IoT Development Tools for PIC32

Presented by:

IoT Development Tools for PIC32
Adios Amigos

Presented by: