CEC Continuing Education Center

DesignNews

Bare Metal C Programming for STM32 Devices

Day 5:

Bare Metal NUCLEO-C071RB :

Doing NUCLEO-F207ZG Things

Sponsored by

DigiKey

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

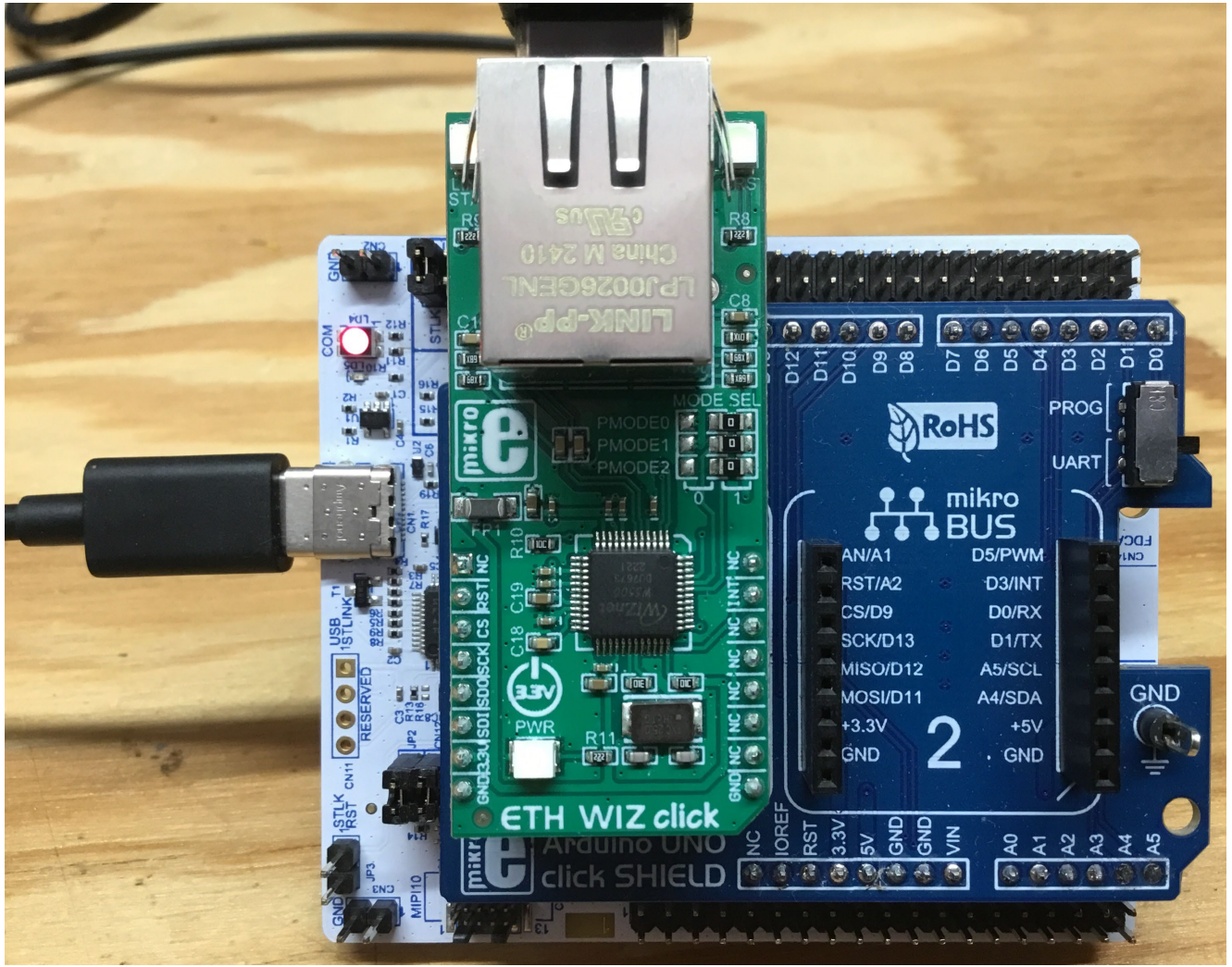
Visit ‘Lecturer Profile’ in your console for more details.

CEC Continuing Education Center

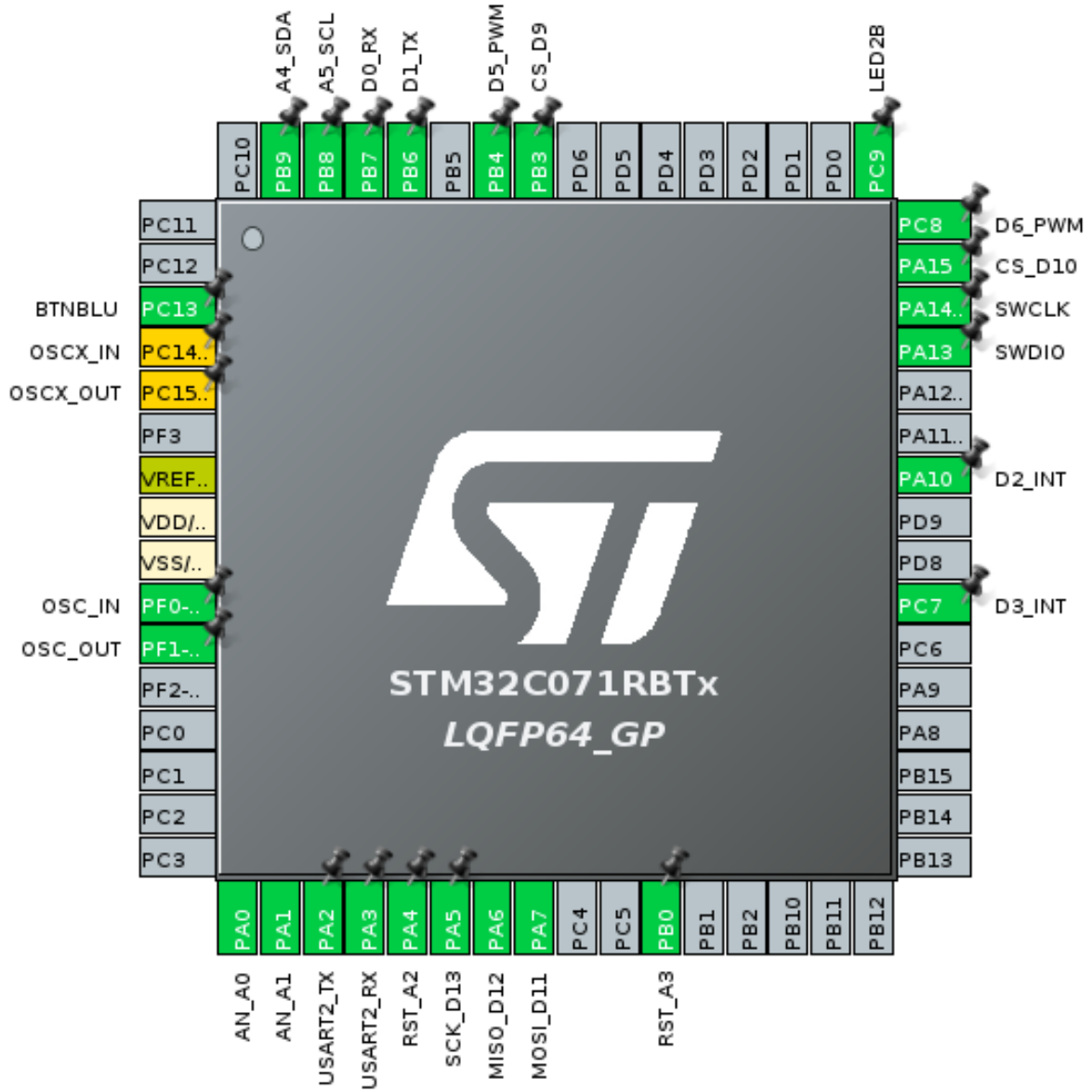
AGENDA

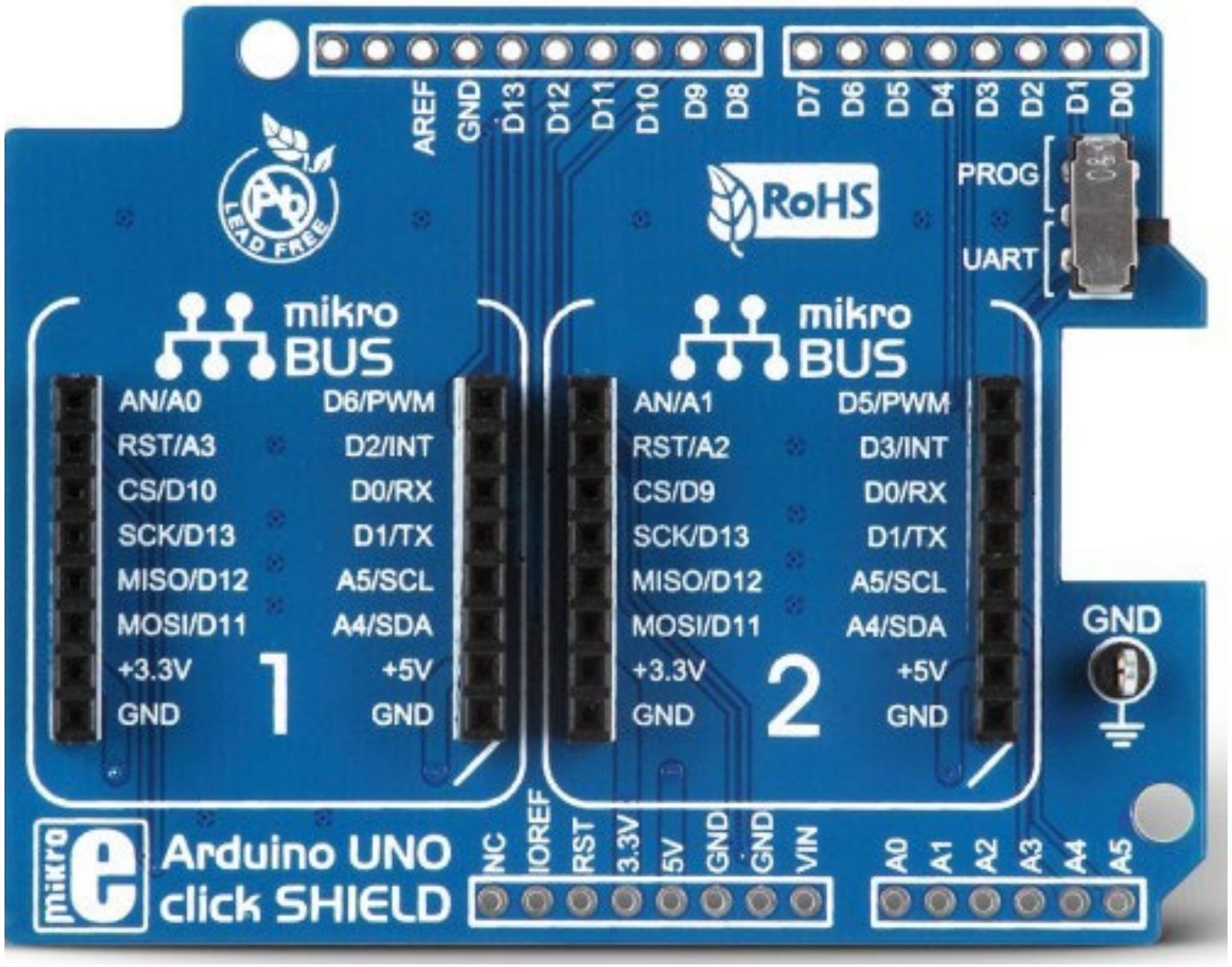
DigiKey

- **Build a VS Code NUCLEO-C071RB ETH WIZ click Application**
- **Build a VS Code NUCLEO-C071RB WizFi360 Application**

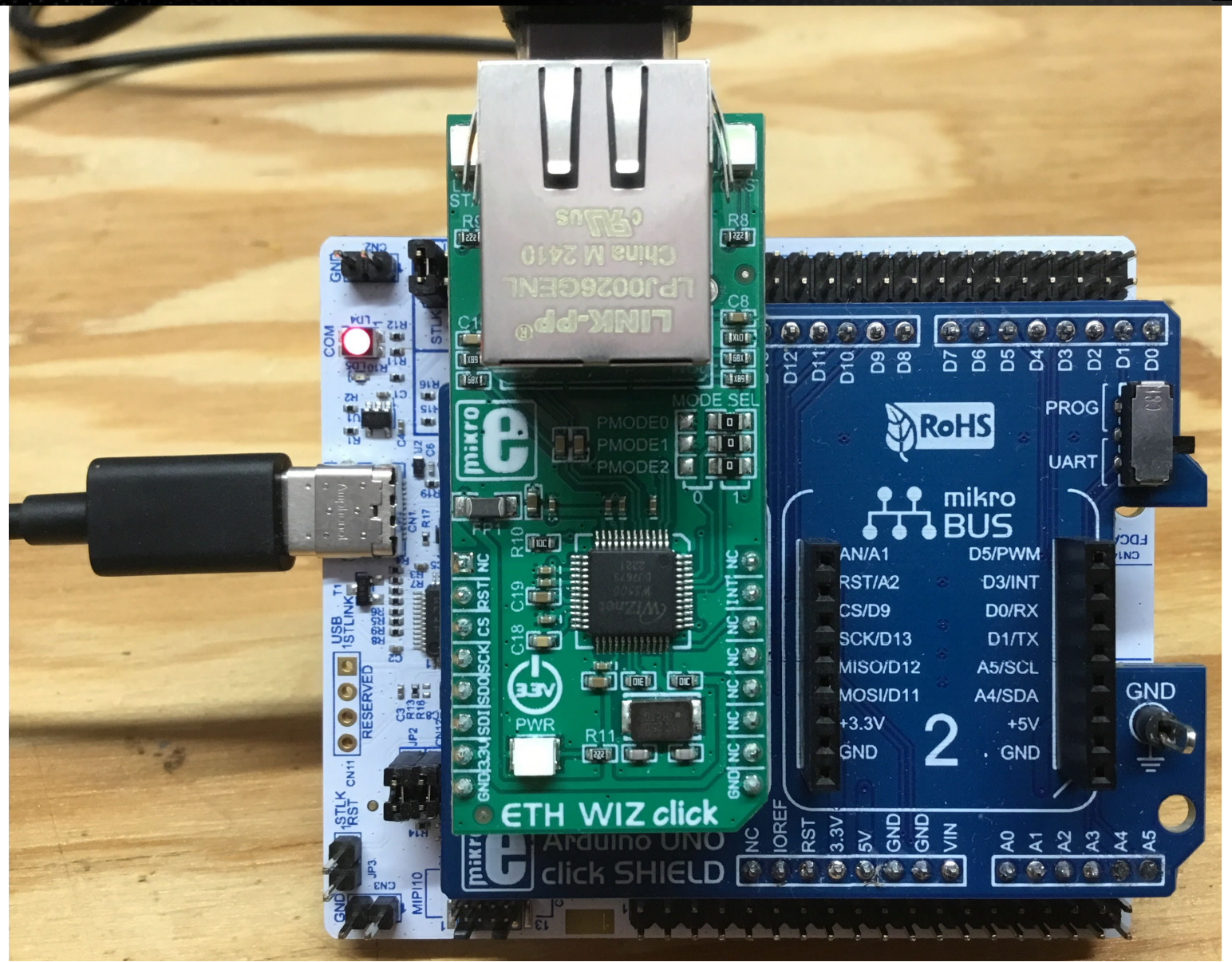


Configure the Hardware





Configure the Hardware

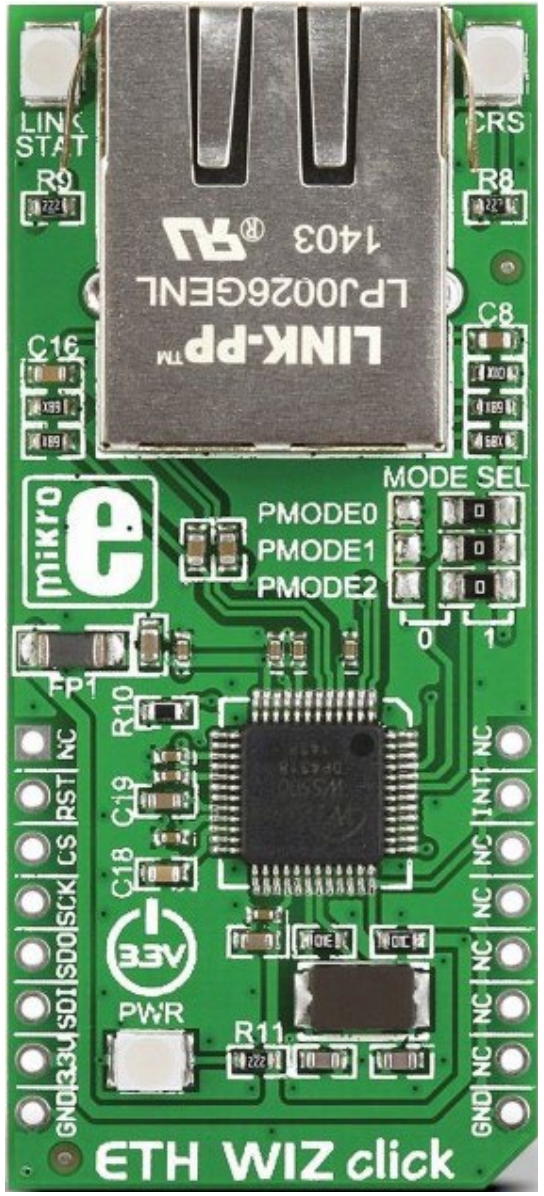


Private Includes and Typedefs

main.c 2

Core > Src > main.c > ...

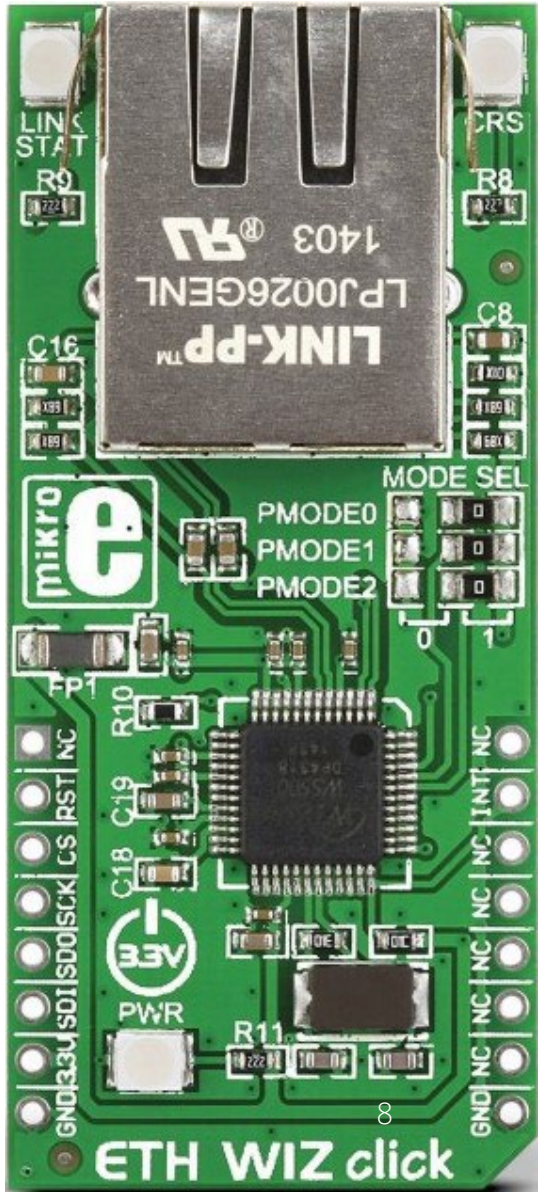
```
13  #include "main.h"
14  #include "adc.h"
15  #include "i2c.h"
16  #include "spi.h"
17  #include "usart.h"
18  #include "gpio.h"
19
20  /* Private includes ----------------------------------------------------------*/
21  /* USER CODE BEGIN Includes */
22  #include <stdio.h>
23  #include "wizchip_conf.h"
24  #include "socket.h"
25  #include "dhcp.h"
26  #include <string.h>
27  #include <stdint.h>
28  /* USER CODE END Includes */
29
30  /* Private typedef -----------------------------------------------------------*/
31  /* USER CODE BEGIN PTD */
32  wiz_NetInfo netInfo = { .mac = {[0]=0x00,[1]=0x04,[2]=0xA3,[3]=0x06,[4]=0xE7,[5]=0x48},
33  |  |  |  |  |  |  .dhcp = NETINFO_DHCP
34  |  |  |  |  |  |  };
35  // pstates
36  enum{
37  |  MODE_DHCP = 0,
38  |  MODE_CONNECT,
39  };
40  /* USER CODE END PTD */
```



Add the User Sources to CMakeLists.txt

CMakeLists.txt

```
45  # Add sources to executable
46  target_sources(${CMAKE_PROJECT_NAME} PRIVATE
47      # Add user sources here
48      /home/fred/nucleo_c071rb_ethwiz/Core/Src/dhcp.c
49      /home/fred/nucleo_c071rb_ethwiz/Core/Src/dns.c
50      /home/fred/nucleo_c071rb_ethwiz/Core/Src/socket.c
51      /home/fred/nucleo_c071rb_ethwiz/Core/Src/w5500.c
52      /home/fred/nucleo_c071rb_ethwiz/Core/Src/wizchip_conf.c
53  )
```

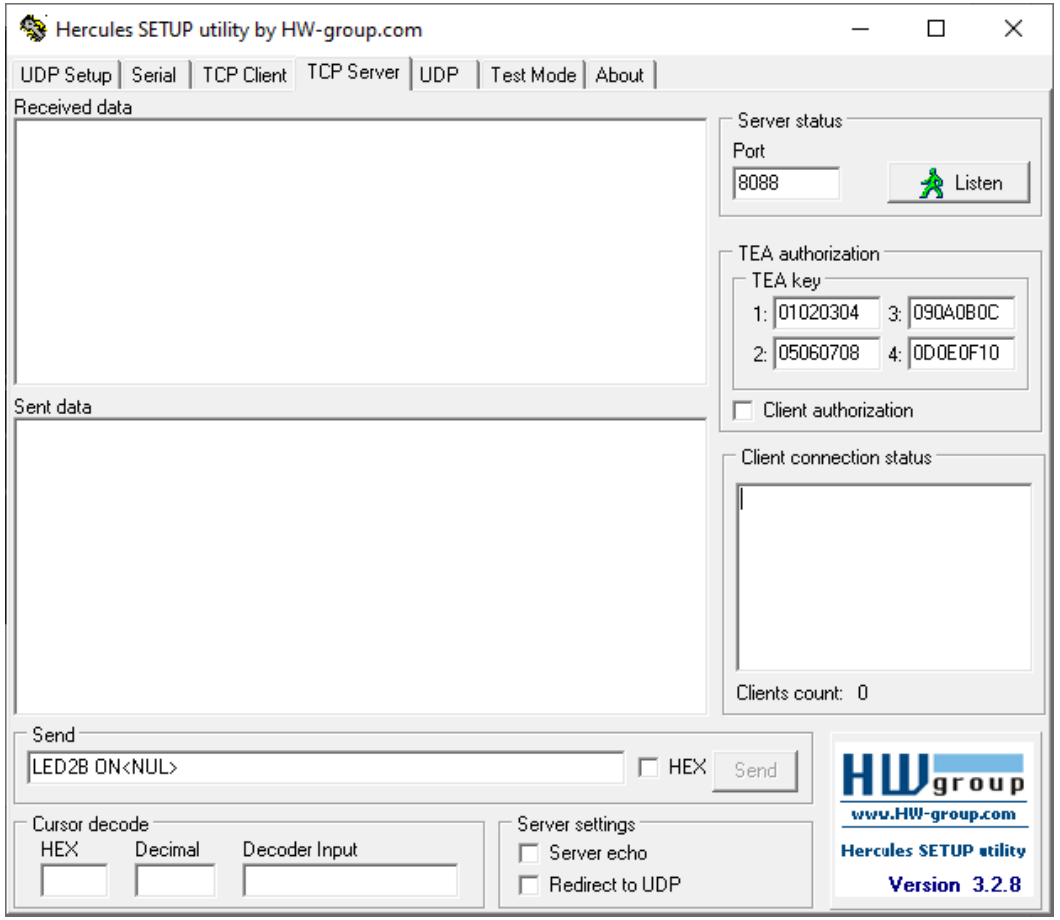


Private User Variables

main.c 2

Core > Src > C main.c > ...

```
69   /* USER CODE BEGIN PV */
70   //*****************************************************************************
71   //* SOCKET USER VARIABLES
72   //*****************************************************************************
73   uint8_t  sktRxBuf[64];
74   uint8_t  retVal;
75   int16_t  rcvLen;
76   uint8_t  bufSize[] = {[0]=2, [1]=2, [2]=2, [3]=2};
77   uint8_t  dhcpBuf[1024];
78   uint8_t  serverIP[] = {[0]=0xC0,[1]=0xA8,[2]=0x01,[3]=0xEB}; //192.168.1.235
79   uint16_t serverPort = 8088;
80   //*****************************************************************************
81   //* USER VARIABLES
82   //*****************************************************************************
83   uint8_t scratch8;
84   uint8_t pstate;
85   uint8_t lastpstate;
86   /* USER CODE END PV */
```

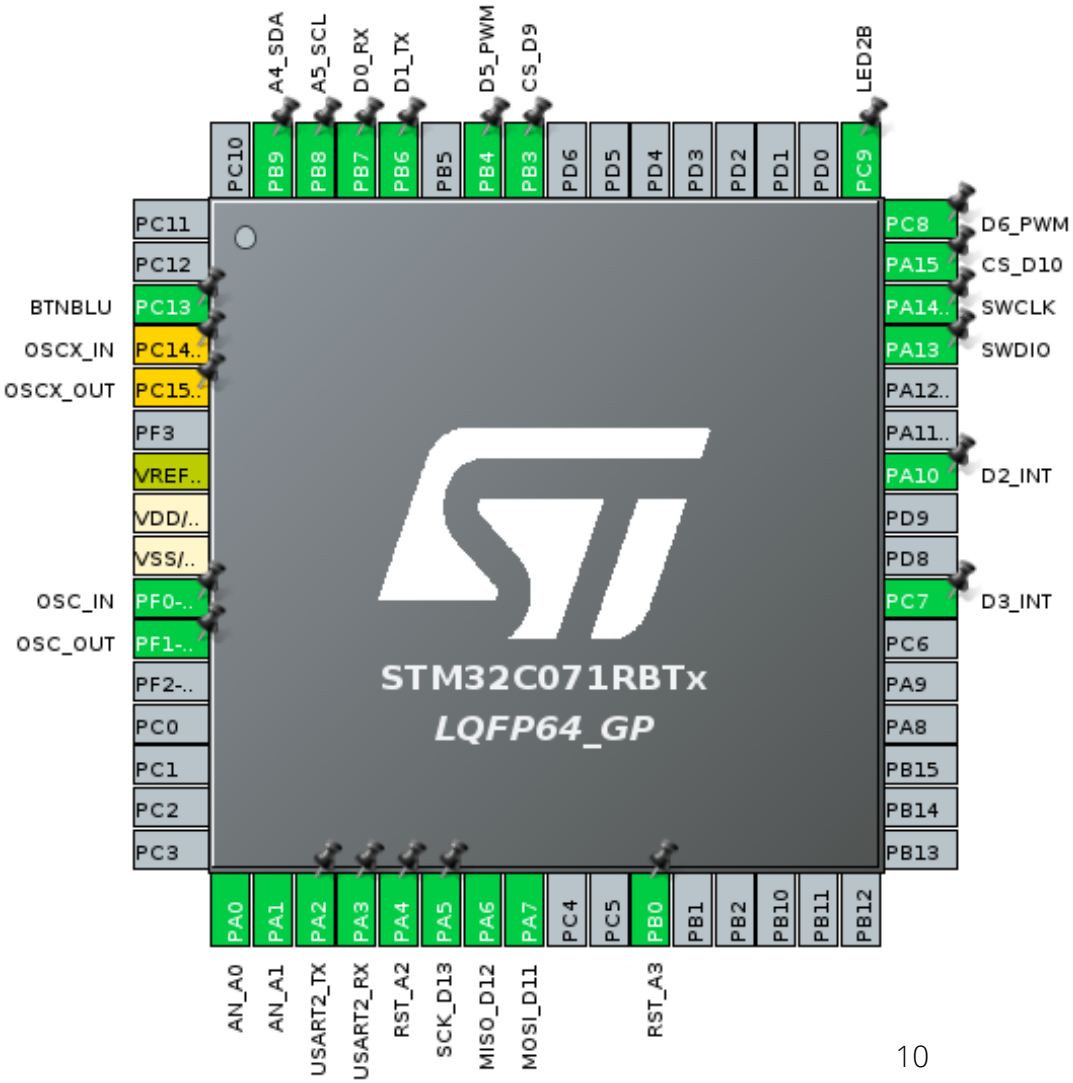


SPI Functions

main.c 2

Core > Src > main.c > ...

```
115  //*****************************************************************************
116  //* WIZNET SPI CS FUNCTIONS
117  //*****************************************************************************
118  void wiz_csLO(void)
119  {
120    HAL_GPIO_WritePin(GPIOx: CS_D10_GPIO_Port, GPIO_Pin: CS_D10_Pin, PinState: GPIO_PIN_RESET);
121  }
122  void wiz_csHI(void)
123  {
124    HAL_GPIO_WritePin(GPIOx: CS_D10_GPIO_Port, GPIO_Pin: CS_D10_Pin, PinState: GPIO_PIN_SET);
125  }
126  //*****************************************************************************
127  //* SPI READ/WRITE FUNCTIONS
128  //*****************************************************************************
129  void ReadBuf(uint8_t* buf, uint16_t len)
130  {
131    HAL_SPI_Receive(hspi: &hspi1,pData: buf,Size: len,Timeout: HAL_MAX_DELAY);
132  }
133  void WriteBuf(uint8_t* buf,uint16_t len)
134  {
135    HAL_SPI_Transmit(hspi: &hspi1,pData: buf,Size: len,Timeout: HAL_MAX_DELAY);
136  }
137  uint8_t ReadByte(void)
138  {
139    uint8_t bite;
140    ReadBuf(buf: &bite,len: sizeof(bite));
141    return bite;
142  }
143  void WriteByte(uint8_t bite)
144  {
145    WriteBuf(buf: &bite,len: sizeof(bite));
146  }
147  /* USER CODE END 0 */
```

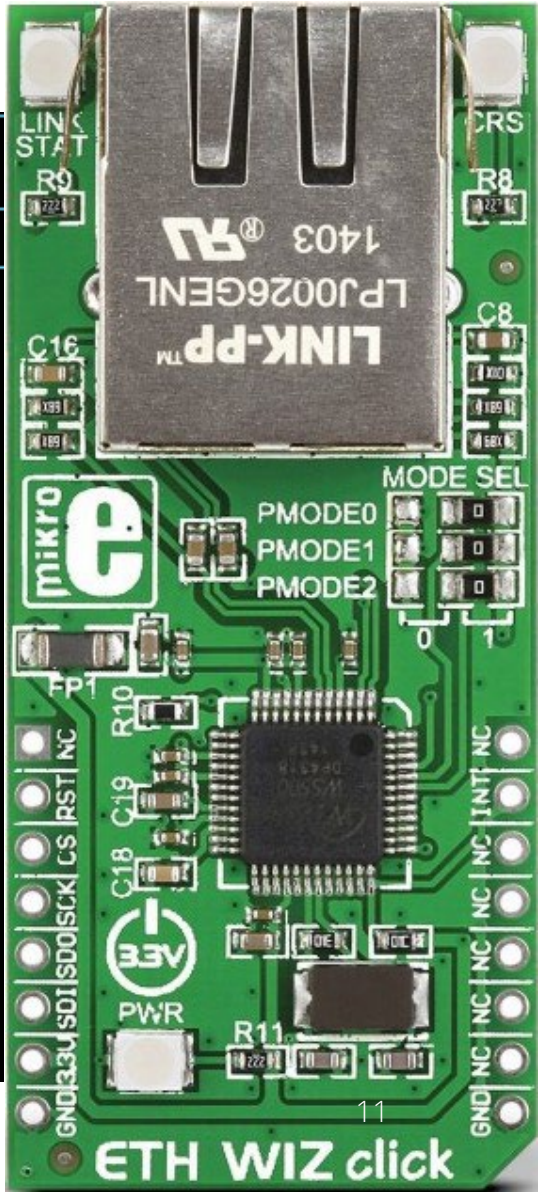


Register Callbacks and DHCP Setup

main.c 2

Core > Src > main.c > main

```
185   /* USER CODE BEGIN 2 */
186
187   pstate = MODE_DHCP;
188   lastpstate = MODE_DHCP;
189
190   printf("INFO: Register Callbacks\r\n");
191   reg_wizchip_cs_cbfunc(cs_sel: wiz_csLO, cs_desel: wiz_csHI);
192   reg_wizchip_spi_cbfunc(spi_rb: ReadByte, spi_wb: WriteByte);
193   reg_wizchip_spiburst_cbfunc(spi_rb: ReadBuf,spi_wb: WriteBuf);
194   printf("INFO: Calling wizchip_init\r\n");
195   wizchip_init(txsize: bufSize, rxsize: bufSize); // Initialize Socket Buffer Size
196   setSHAR(pBuf: netInfo.mac); // Set the local MAC Address
197
198   printf("INFO: DHCP_Init\r\n");
199   DHCP_init(s: 0,buf: dhcpBuf);
200   /* USER CODE END 2 */
```

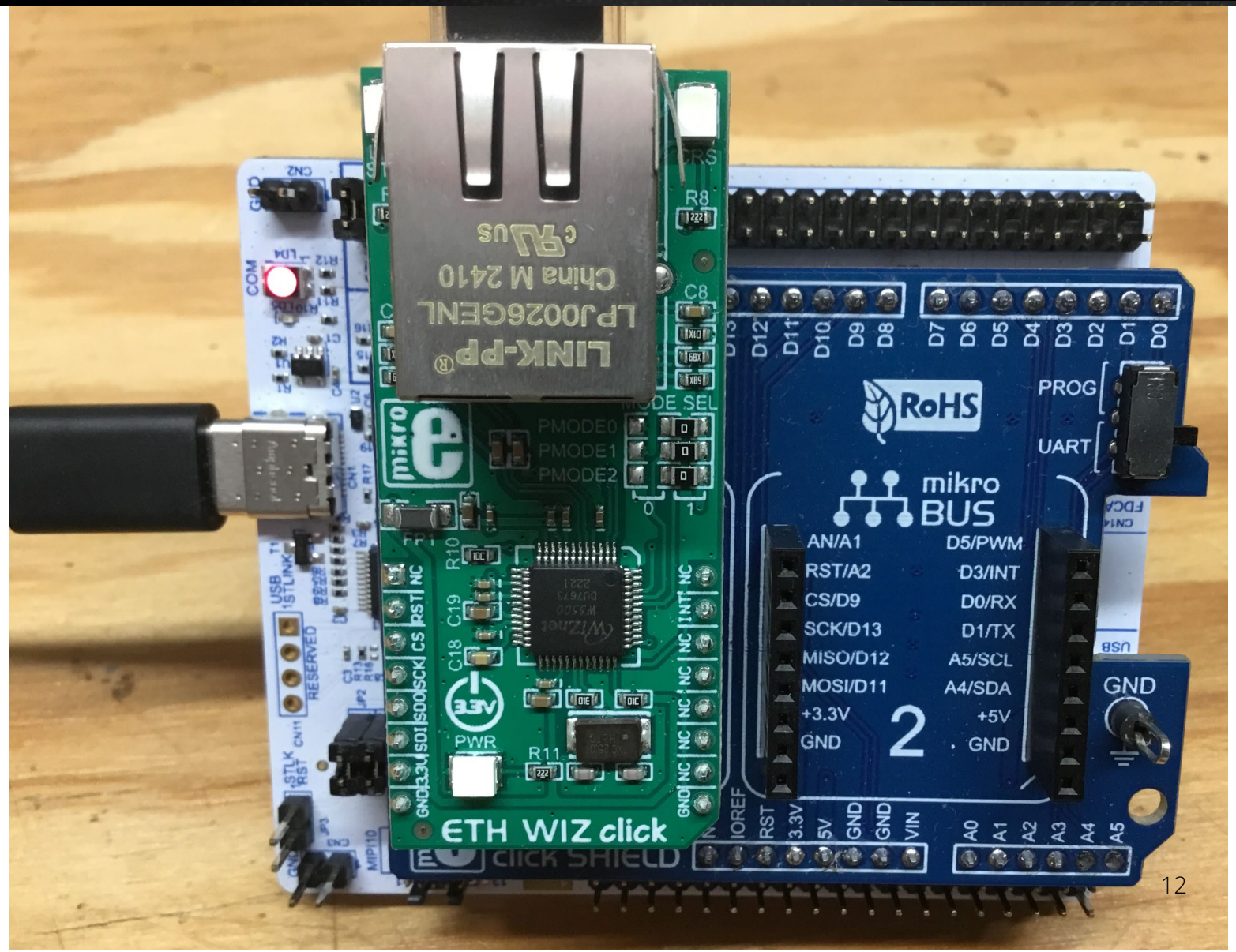


DHCP Module

main.c 2

Core > Src > main.c > main

```
203   /* USER CODE BEGIN WHILE */
204   while (1)
205   {
206     switch(pstate)
207     {
208       //*****************************************
209       //* MODE_DHCP
210       //*****************************************
211       case MODE_DHCP:
212       switch(DHCP_run())
213       {
214         case DHCP_FAILED:
215           printf("DHCP FAILED\r\n");
216         break;
217         case DHCP_RUNNING:
218         break;
219         case DHCP_IP_ASSIGN:
220           printf("INFO: DHCP IP ASSIGN\r\n");
221         break;
222         case DHCP_IP_CHANGED:
223           printf("INFO: DHCP IP CHANGED\r\n");
224         break;
225         case DHCP_IP_LEASED:
226           printf("INFO: DHCP IP LEASED\r\n");
227           wizchip_getnetinfo(pnetinfo: &netInfo);
228           showNetInfo();
229           wizchip_setnetinfo(pnetinfo: &netInfo);
230           pstate = MODE_CONNECT;
231         break;
232         case DHCP_STOPPED:
233           printf("INFO: DHCP STOPPED\r\n");
234         break;
235       }
236       break;
```

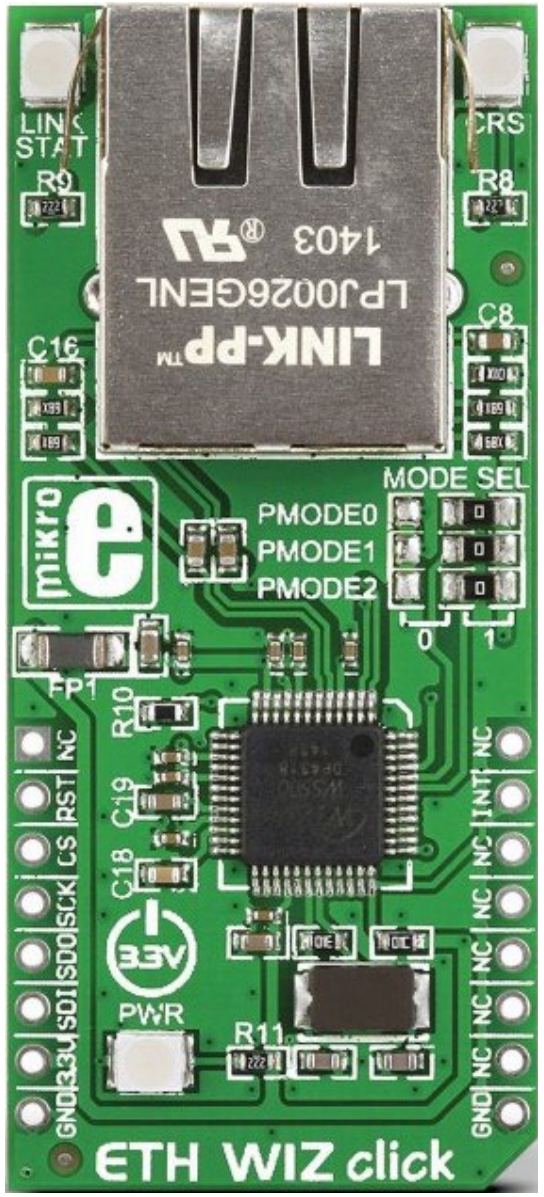


Connect Module

main.c 2

Core > Src > main.c > main

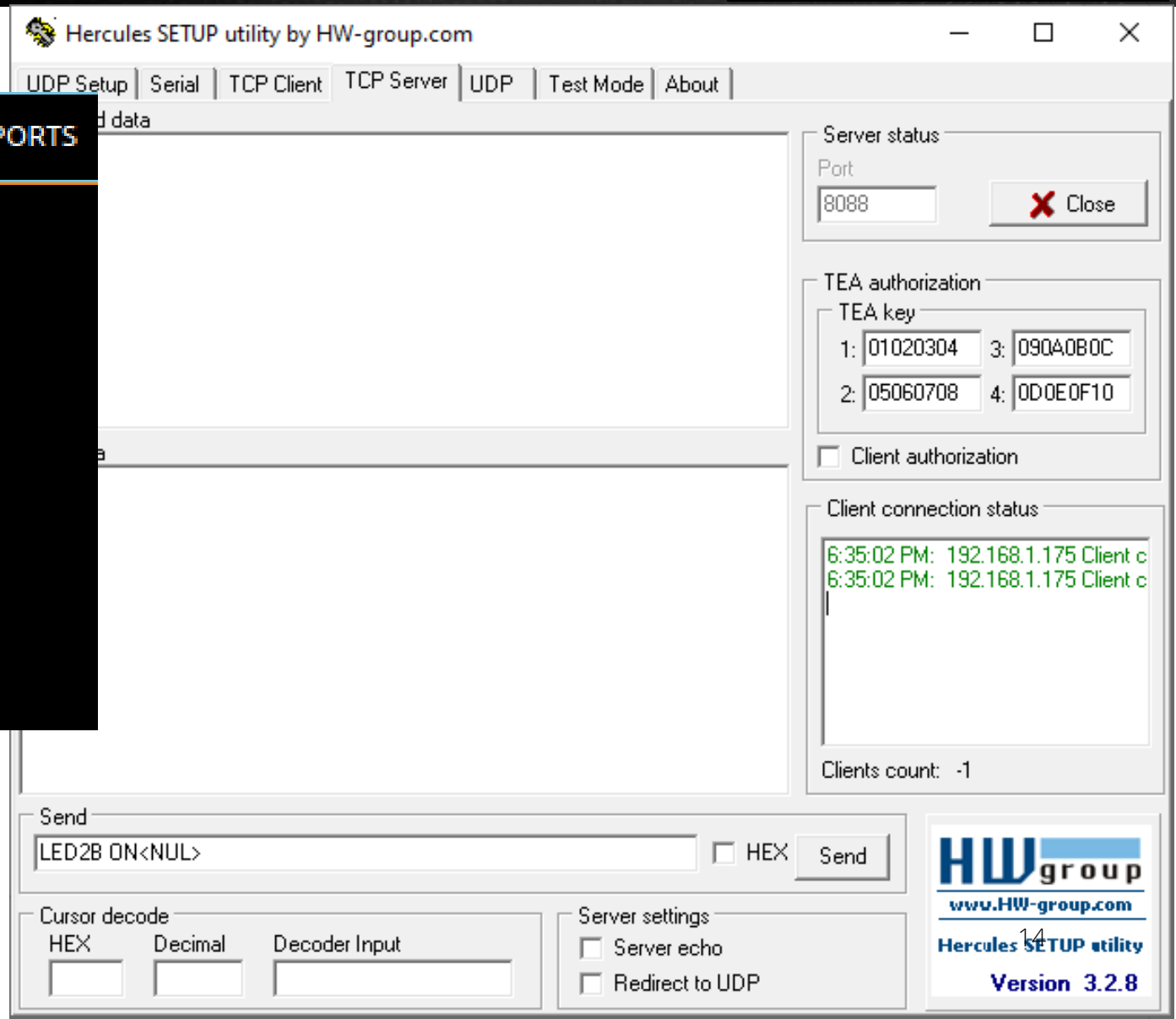
```
238         //*****************************************************************************
239         //* MODE_CONNECT
240         //*****************************************************************************
241         case MODE_CONNECT:
242           retVal = socket(sn: 0, protocol: Sn_MR_TCP, port: 5000, flag: 0);
243           if(retVal == 0)
244           {
245             printf("INFO: Socket 0 opened.\r\n");
246             retVal = connect(sn: 0, addr: serverIP, port: 8088);
247             if(retVal == SOCK_OK)
248             {
249               printf("INFO: Connected\r\n");
250               HAL_Delay(Delay: 2000);
251               do{
252                 rcvLen = 0;
253                 do{
254                   rcvLen = getSn_RX_RSR(sn: 0);
255                 }while(rcvLen == 0);
256                 recv(sn: 0, buf: sktRxBuf, len: rcvLen);
257                 //HAL_UART_Transmit(&huart2, sktRxBuf, rcvLen, HAL_MAX_DELAY);
258                 if(strncmp(sktRxBuf,"LED2B ON",9) == 0)
259                 {
260                   HAL_GPIO_WritePin(GPIOx: LED2B_GPIO_Port, GPIO_Pin: LED2B_Pin, PinState: GPIO_PIN_RESET);
261                   printf("INFO: BLUE LED ON!\r\n");
262                 }
263                 else if(strncmp(sktRxBuf,"LED2B OFF",10) == 0)
264                 {
265                   HAL_GPIO_WritePin(GPIOx: LED2B_GPIO_Port, GPIO_Pin: LED2B_Pin, PinState: GPIO_PIN_SET);
266                   printf("INFO: BLUE LED OFF!\r\n");
267                 }
268               }while(1);
269             }
270           }
271         break;
272       }
```



Turn MAX Loose

```
PROBLEMS 7   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

INFO: Register Callbacks
INFO: Calling wizchip_init
INFO: DHCP_Init
INFO: DHCP IP LEASED
Network configuration:
  IP ADDRESS:  192.168.1.175
  MAC ADDRESS: 0x00:0x04:0xA3:0x06:0xE7:0x48
  NETMASK:     255.255.255.0
  GATEWAY:     192.168.1.1
  DNS:         0.0.0.0
INFO: Socket 0 opened.
INFO: Connected
```



Turn MAX Loose

```
PROBLEMS 7    OUTPUT    DEBUG CONSOLE    TERMINAL    P
INFO: Register Callbacks
INFO: Calling wizchip_init
INFO: DHCP_Init
INFO: DHCP IP LEASED
Network configuration:
   IP ADDRESS:   192.168.1.175
   MAC ADDRESS: 0x00:0x04:0xA3:0x06:0xE7:0x48
   NETMASK:      255.255.255.0
   GATEWAY:      192.168.1.1
   DNS:          0.0.0.0
INFO: Socket 0 opened.
INFO: Connected
INFO: BLUE LED ON!
```

Hercules SETUP utility by HW-group.com

UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About

Received data

Sent data

LED2B ON{00}

Server status
Port 8088 Close

TEA authorization
TEA key
1: 01020304 3: 090A0B0C
2: 05060708 4: 0D0E0F10
Client authorization

Client connection status
6:36:30 PM: 192.168.1.175 Client c
6:36:31 PM: 192.168.1.175 Client c
Clients count: -2

Send
LED2B ON<NUL> HEX Send

Cursor decode
HEX Decimal Decoder Input

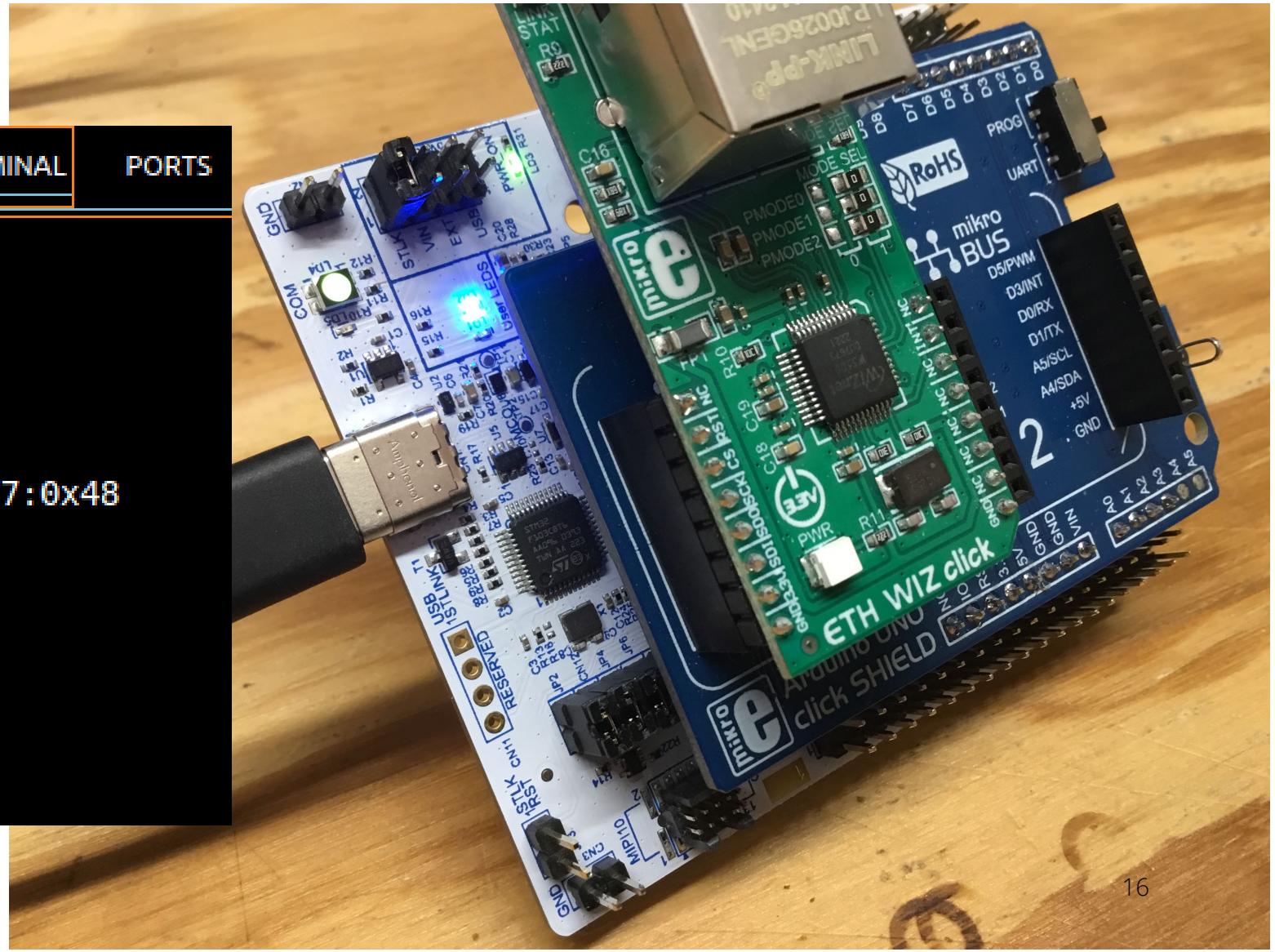
Server settings
Server echo
Redirect to UDP

HW group www.HW-group.com Hercules SETUP utility Version 3.2.8

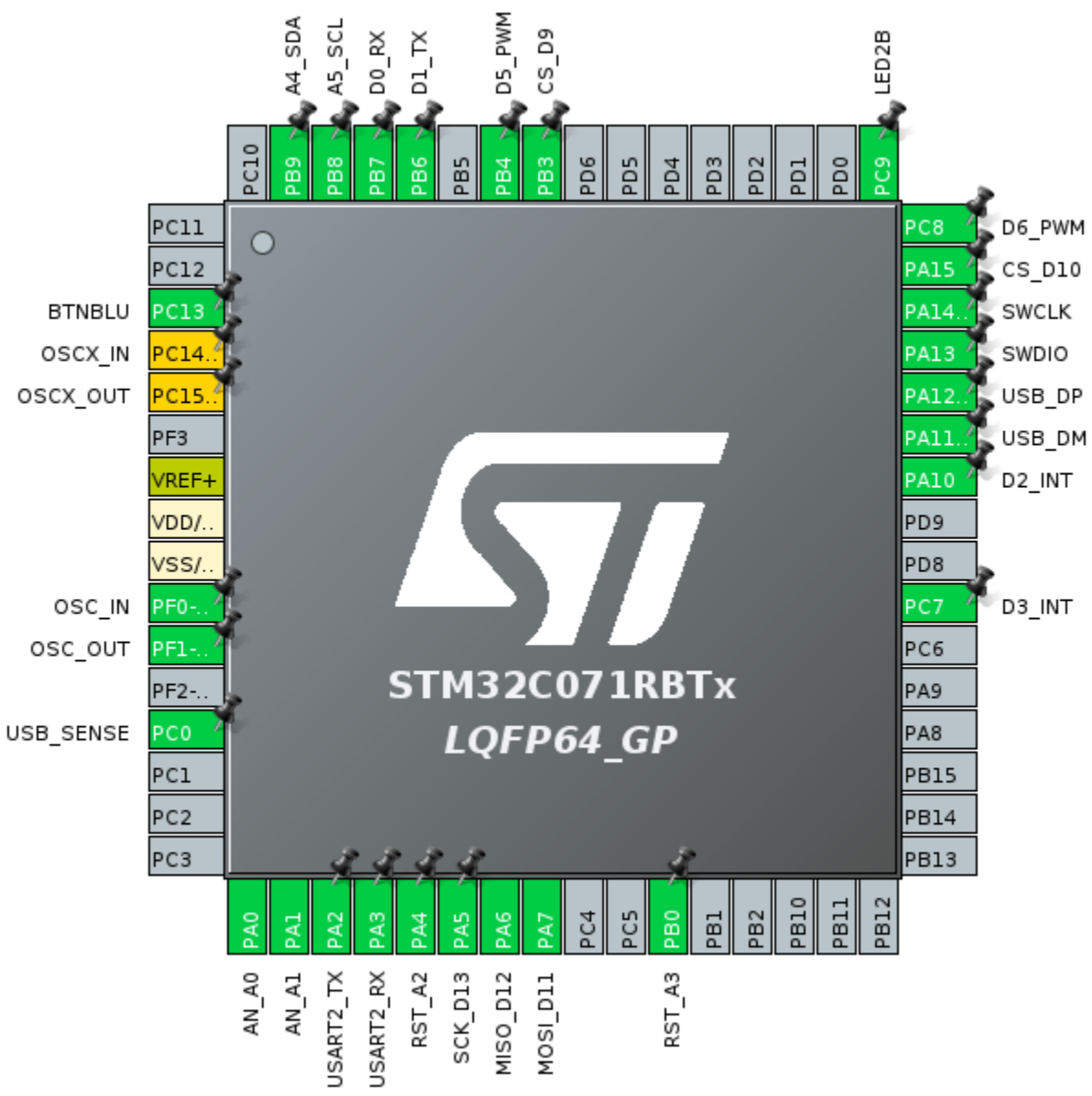
Turn MAX Loose

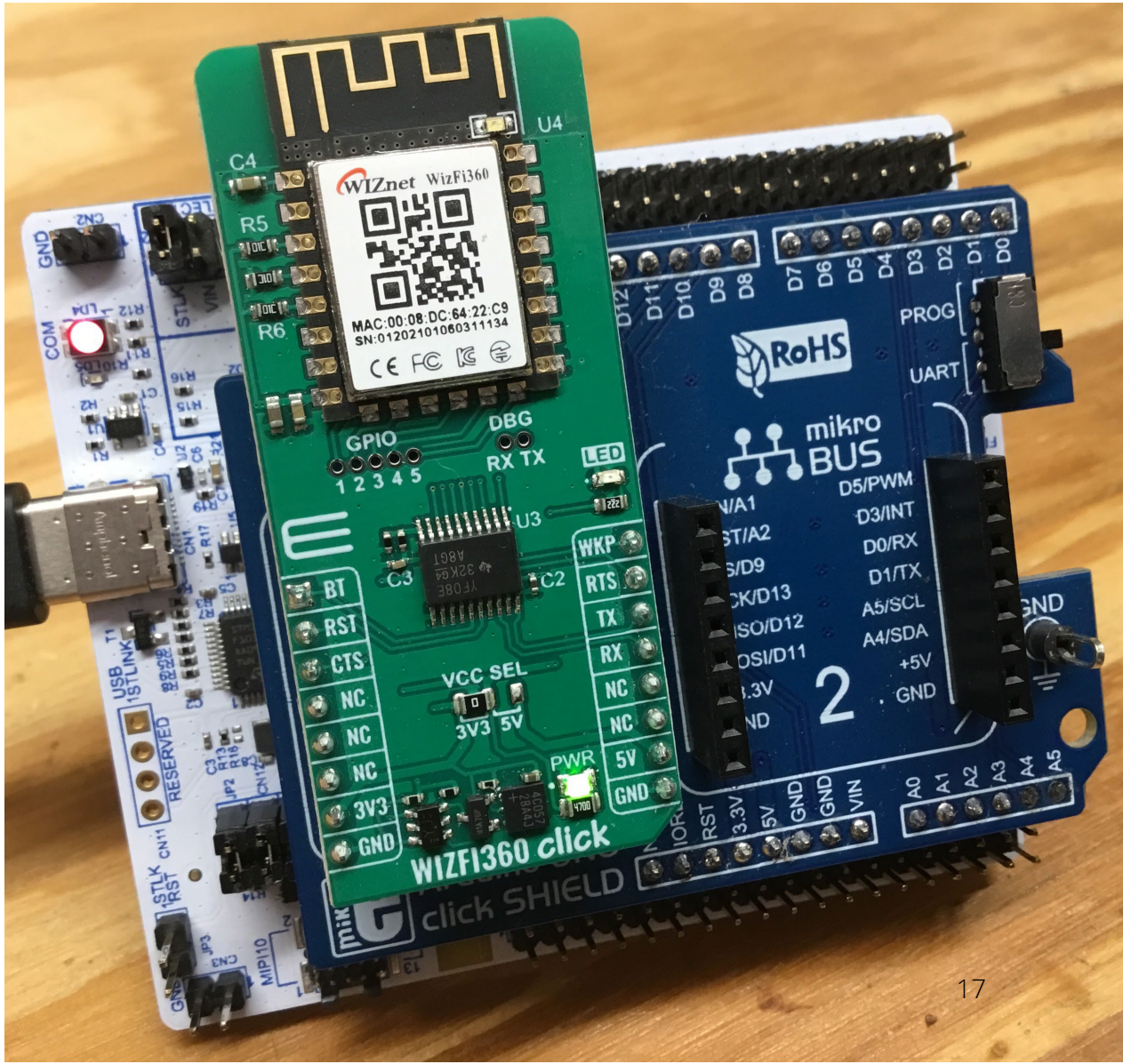
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
INFO: Register Callbacks
INFO: Calling wizchip_init
INFO: DHCP_Init
INFO: DHCP IP LEASED
Network configuration:
   IP ADDRESS:   192.168.1.175
   MAC ADDRESS:  0x00:0x04:0xA3:0x06:0xE7:0x48
   NETMASK:      255.255.255.0
   GATEWAY:      192.168.1.1
   DNS:          0.0.0.0
INFO: Socket 0 opened.
INFO: Connected
INFO: BLUE LED ON!
```



Configure the Hardware



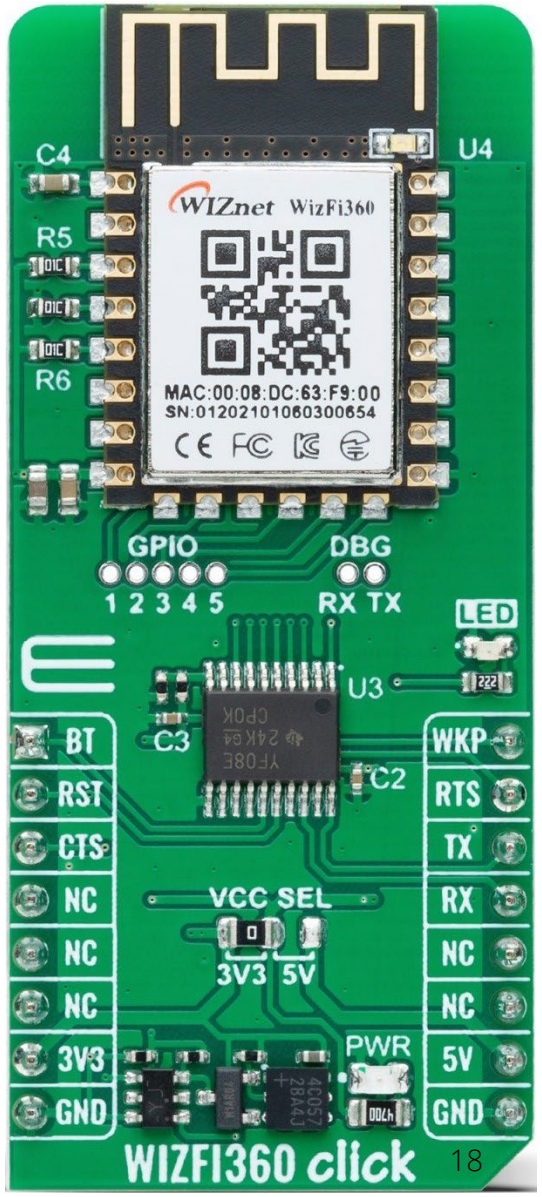


AT Functions

main.c 9

Core > Src > main.c > chk_atok

```
75  /* USER CODE BEGIN PFP */
76  void errorBlink(void);
77  void hardReset360(void);
78  void sendAT(void);
79  uint8_t chk_atok(void);
80  uint8_t setStationMode(void);
81  uint8_t setSingleConnectionMode(void);
82  uint8_t setDhcpEnable(void);
83  uint8_t connect2AP(void);
84  uint8_t connect2Server(void);
85  uint8_t sendLedON(void);
86  uint8_t sendLedOFF(void);
87  uint8_t sendReady(void);
88  /* USER CODE END PFP */
89
90  /* Private user code ---------------------------------------------------------*/
91  /* USER CODE BEGIN 0 */
92  //*****************************************************************************
93  //* SEND AT
94  //*****************************************************************************
95  void sendAT(void)
96  {
97      uint8_t cmd[]={[0]='A',[1]='T',[2]='\r',[3]='\n'};
98      HAL_UART_Transmit(huart: &huart1, pData: cmd, Size: sizeof(cmd), Timeout: HAL_MAX_DELAY);
99  }
```

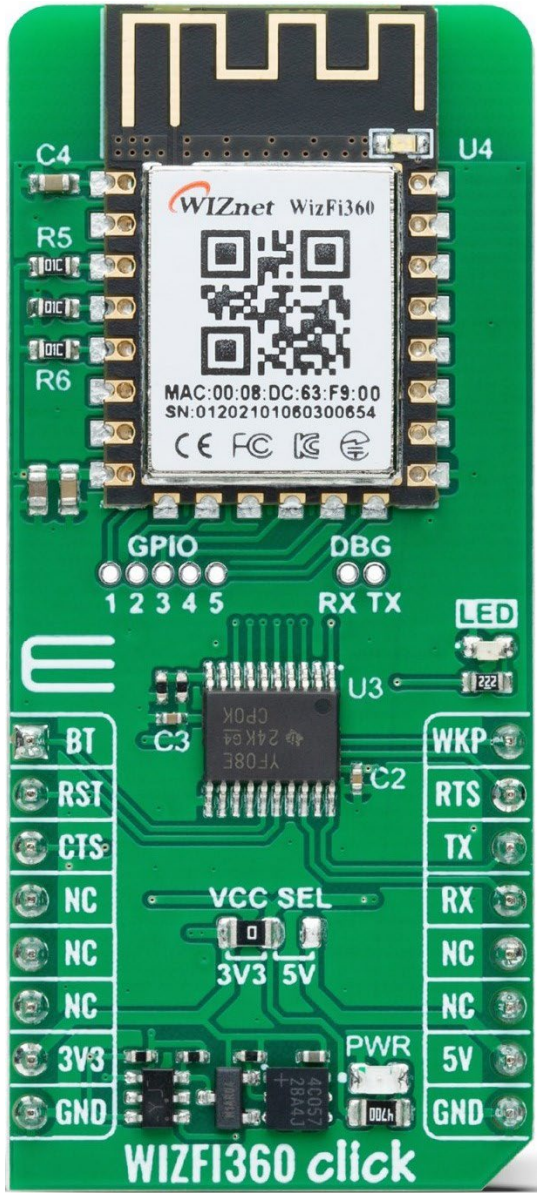


Initialization Functions

main.c 9

Core > Src > C main.c > ...

```
400   //*****************************************************************************
401   //* SEND INITIAL AT COMMAND
402   //*****************************************************************************
403   okat = 0;
404   RingBuffer_Init();
405   hardReset360();
406   resetRxRingBuf();
407   do
408   {
409       resetRxRingBuf();
410       rxBufIndx = 0x00;
411       sendAT();
412         HAL_Delay(Delay: 50);
413         do
414         {
415           if(IsCharInQueue())
416           {
417               rxBuf[rxBufIndx++] = readRxRingBuf();
418           }
419         }while(IsCharInQueue());
420         if(rxBuf[rxBufIndx-4] == 'O' && rxBuf[rxBufIndx-3] == 'K')
421           {
422             okat = 0x01;
423           }
424   }while(okat == 0);
```

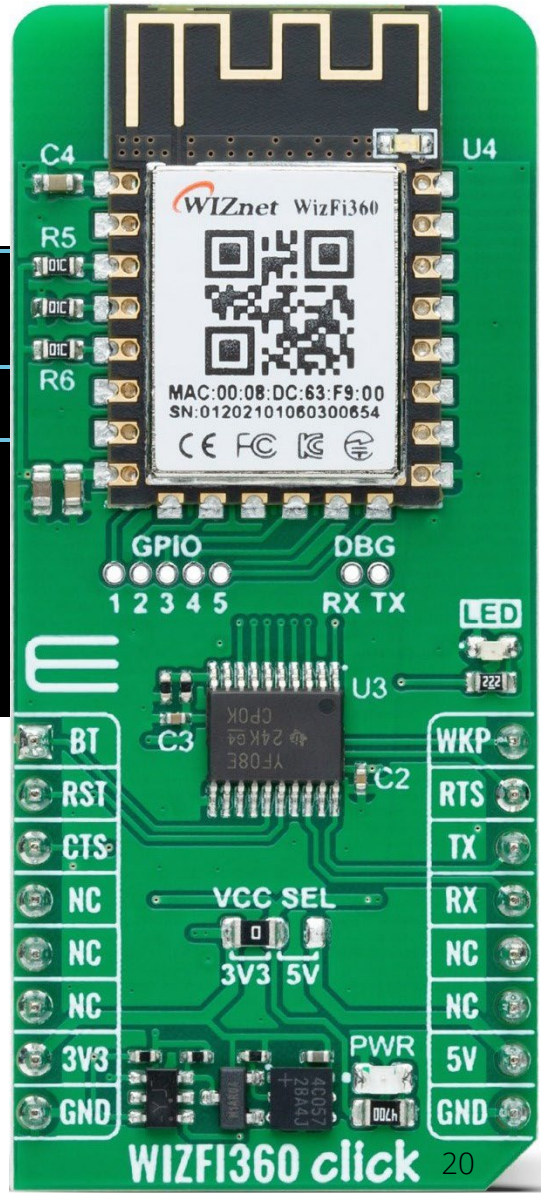


Add STM32RingBuffer.c to the Project

CMakeLists.txt

CMakeLists.txt

```
46  target_sources(${CMAKE_PROJECT_NAME} PRIVATE
47      # Add user sources here
48      /home/fred/nucleo_c071rb_wizfi360/Core/Src/STM32RingBuffer.c
49  )
```

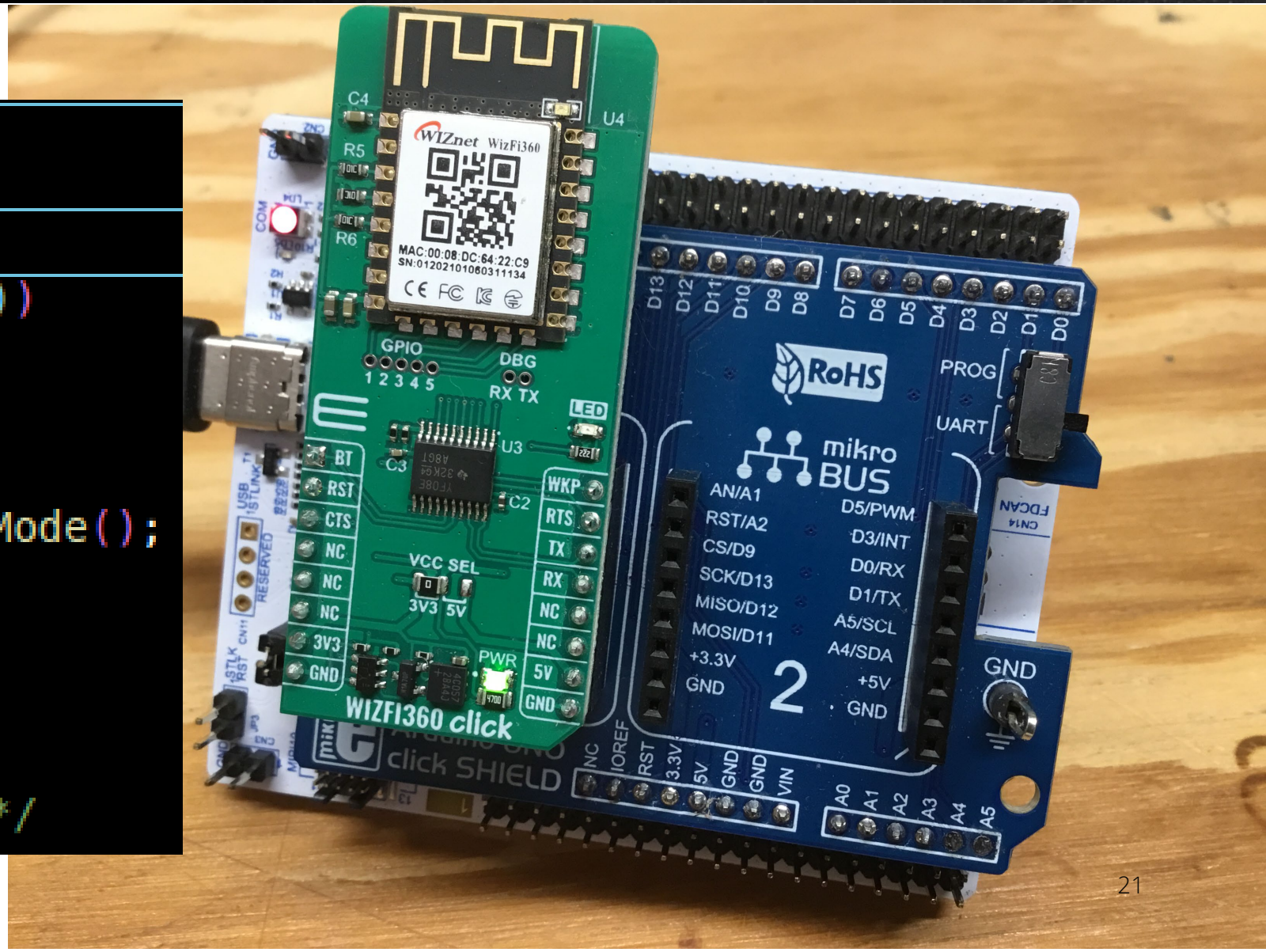


Application Main Flow

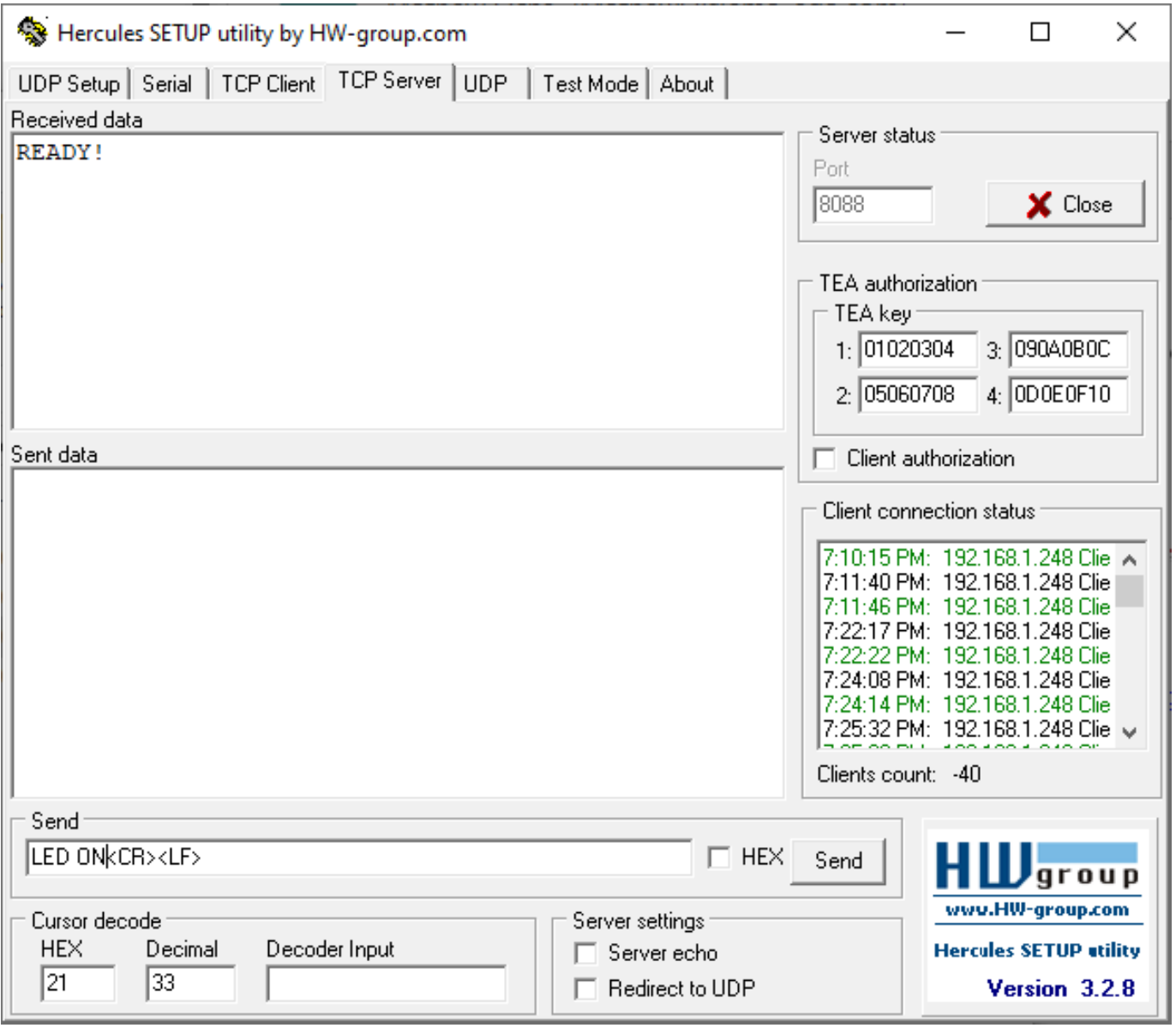
main.c 9

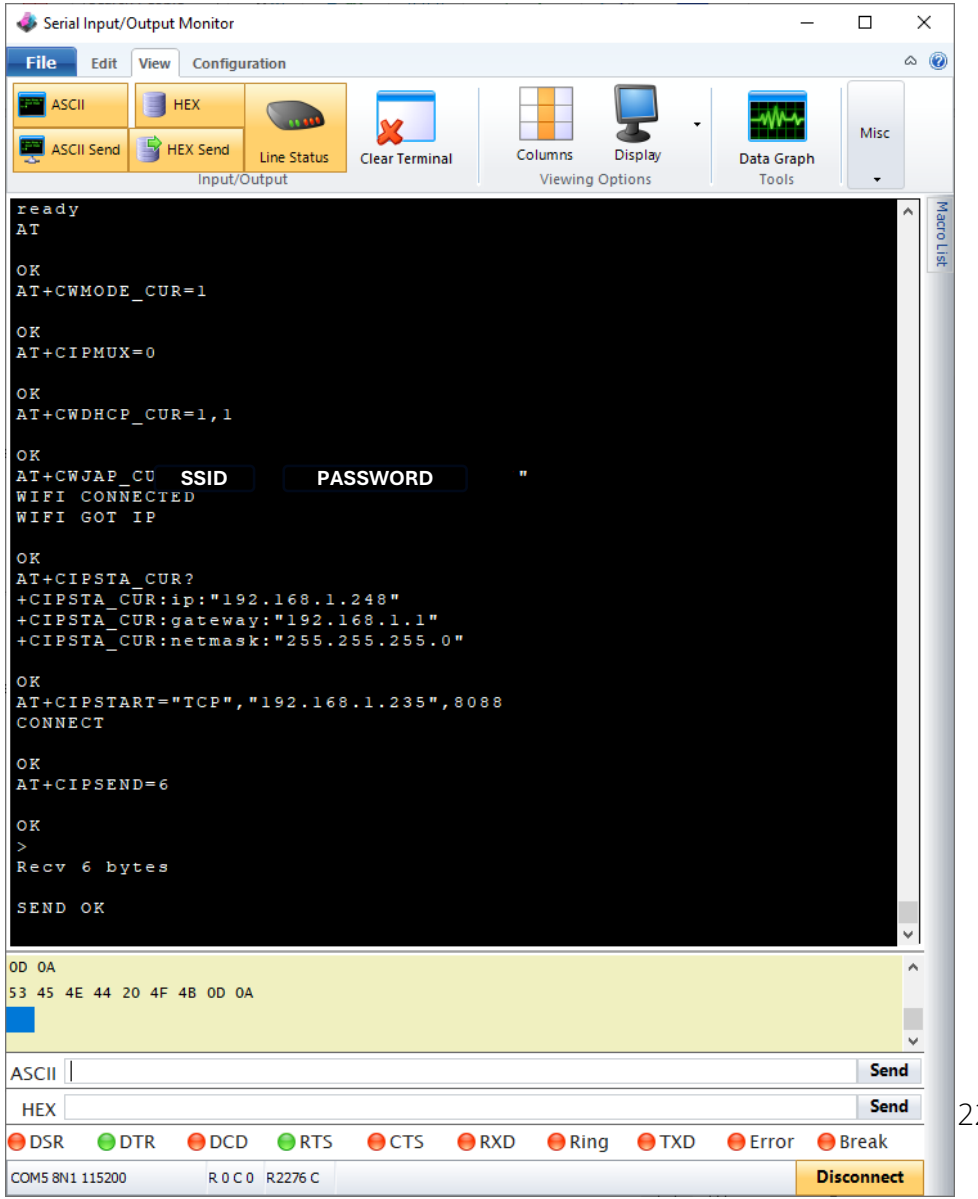
Core > Src > main.c > ...

```
426   if(!setStationMode())
427   {
428     errorBlink();
429   }
430   setSingleConnectionMode();
431   setDhcpEnable();
432   connect2AP();
433   connect2Server();
434   sendReady();
435   /* USER CODE END 2 */
```

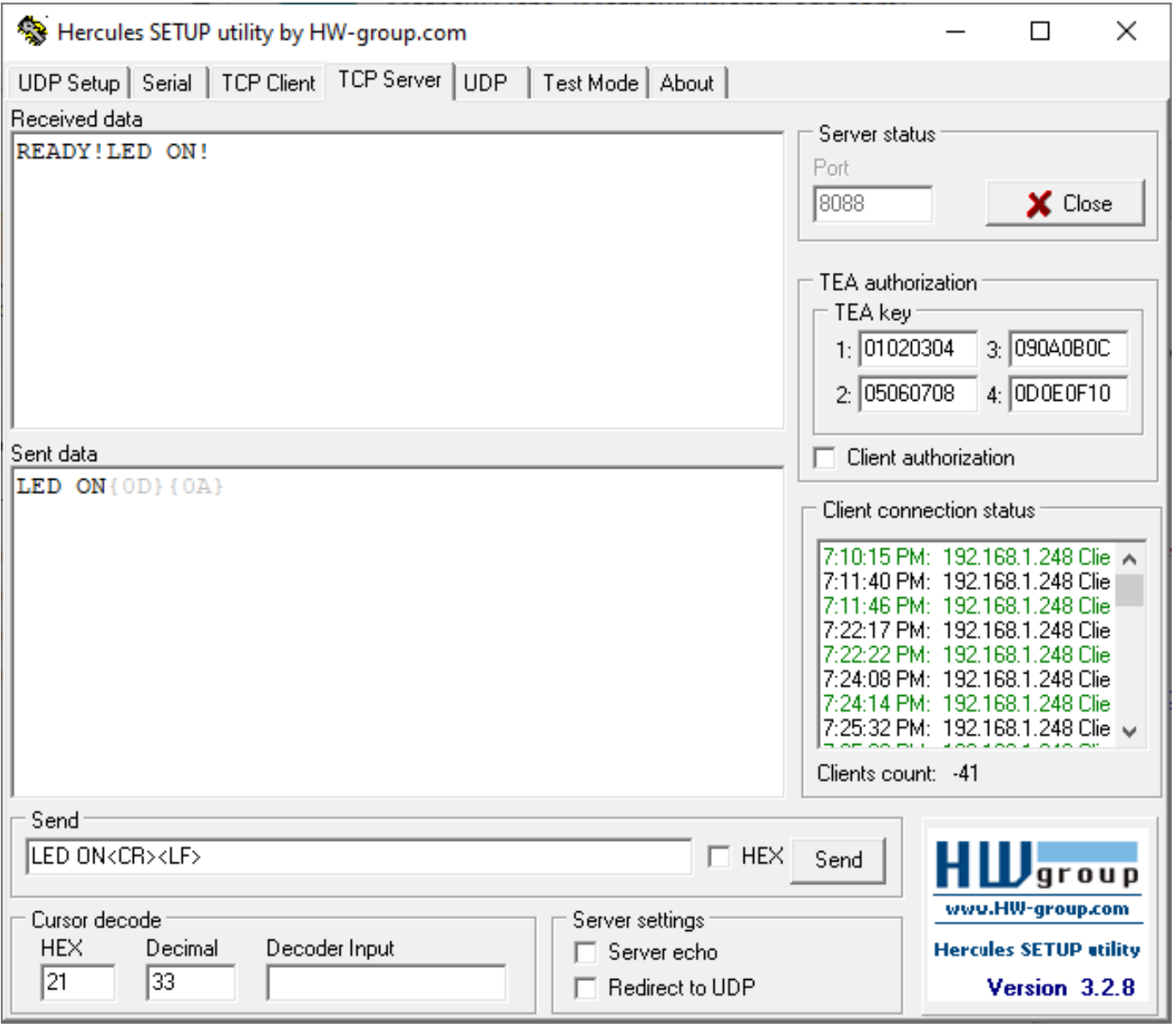


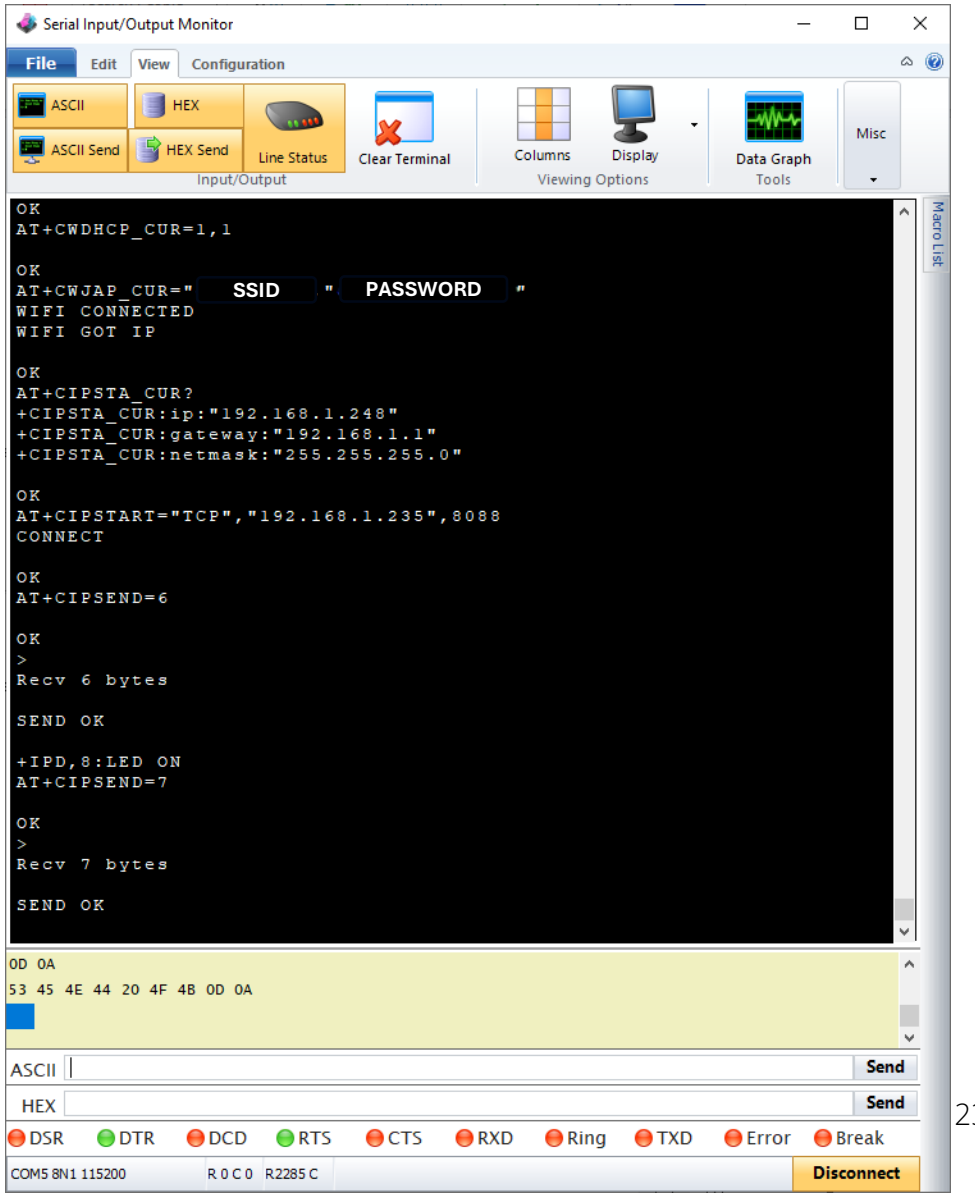
Connect to the Server





Send the LED ON Command





Send the LED ON Command

Hercules SETUP utility by HW-group.com

UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About

Received data

```
READY!LED ON!
```

Server status

Port: 8088 — Close

TEA authorization

TEA key

1: 01020304 3: 090A0B0C

2: 05060708 4: 0D0E0F10

Client authorization

Sent data

```
LED ON{0D}{0A}
```

Client connection status

```
7:10:15 PM:  192.168.1.248 Clie
7:11:40 PM:  192.168.1.248 Clie
7:11:46 PM:  192.168.1.248 Clie
7:22:17 PM:  192.168.1.248 Clie
7:22:22 PM:  192.168.1.248 Clie
7:24:08 PM:  192.168.1.248 Clie
7:24:14 PM:  192.168.1.248 Clie
7:25:32 PM:  192.168.1.248 Clie
```

Clients count: -41

Send

LED ON<CR><LF> — HEX — Send

Cursor decode

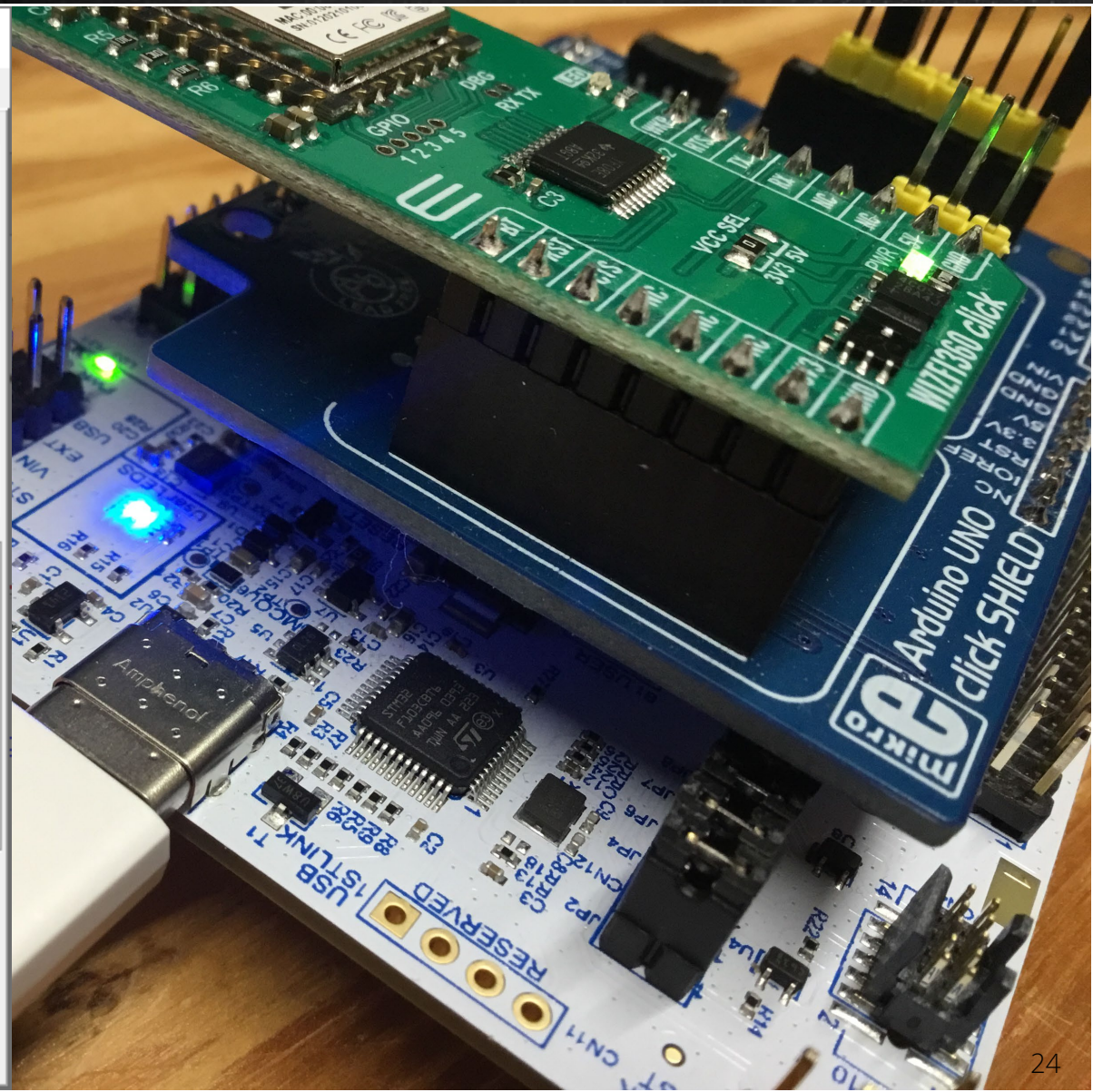
HEX: 21 Decimal: 33 Decoder Input:

Server settings

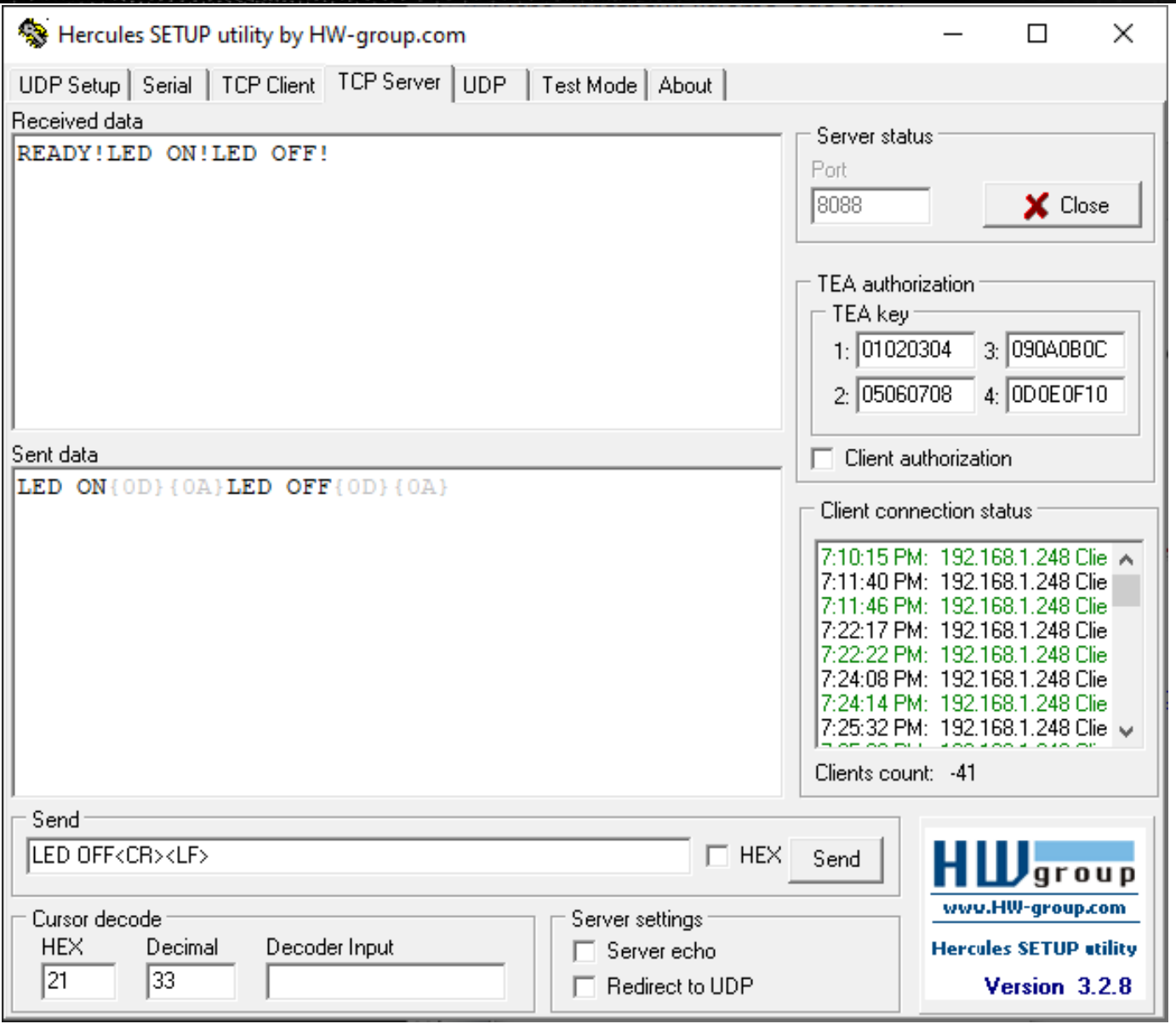
Server echo

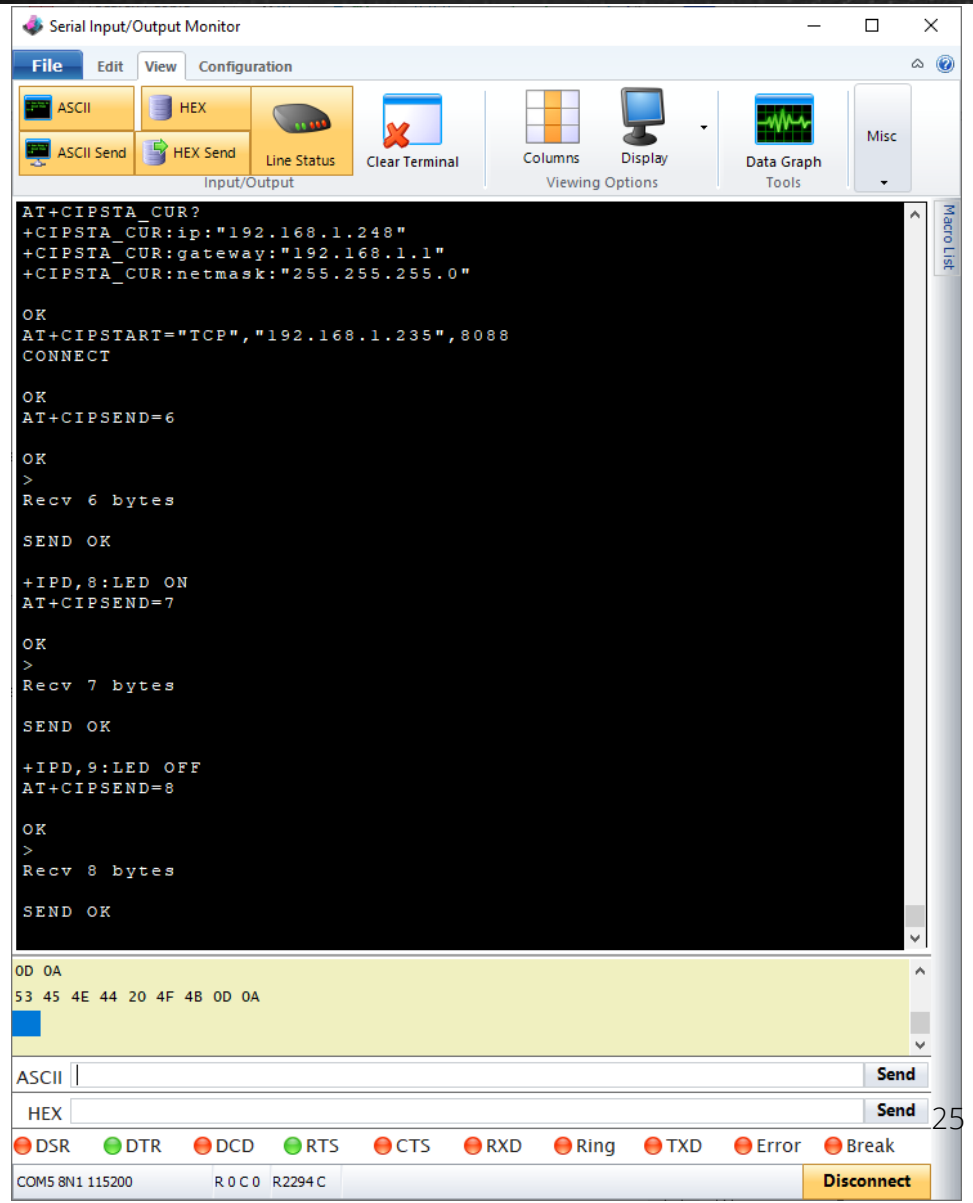
Redirect to UDP

HW group
www.HW-group.com
Hercules SETUP utility
Version 3.2.8



Send the LED OFF Command



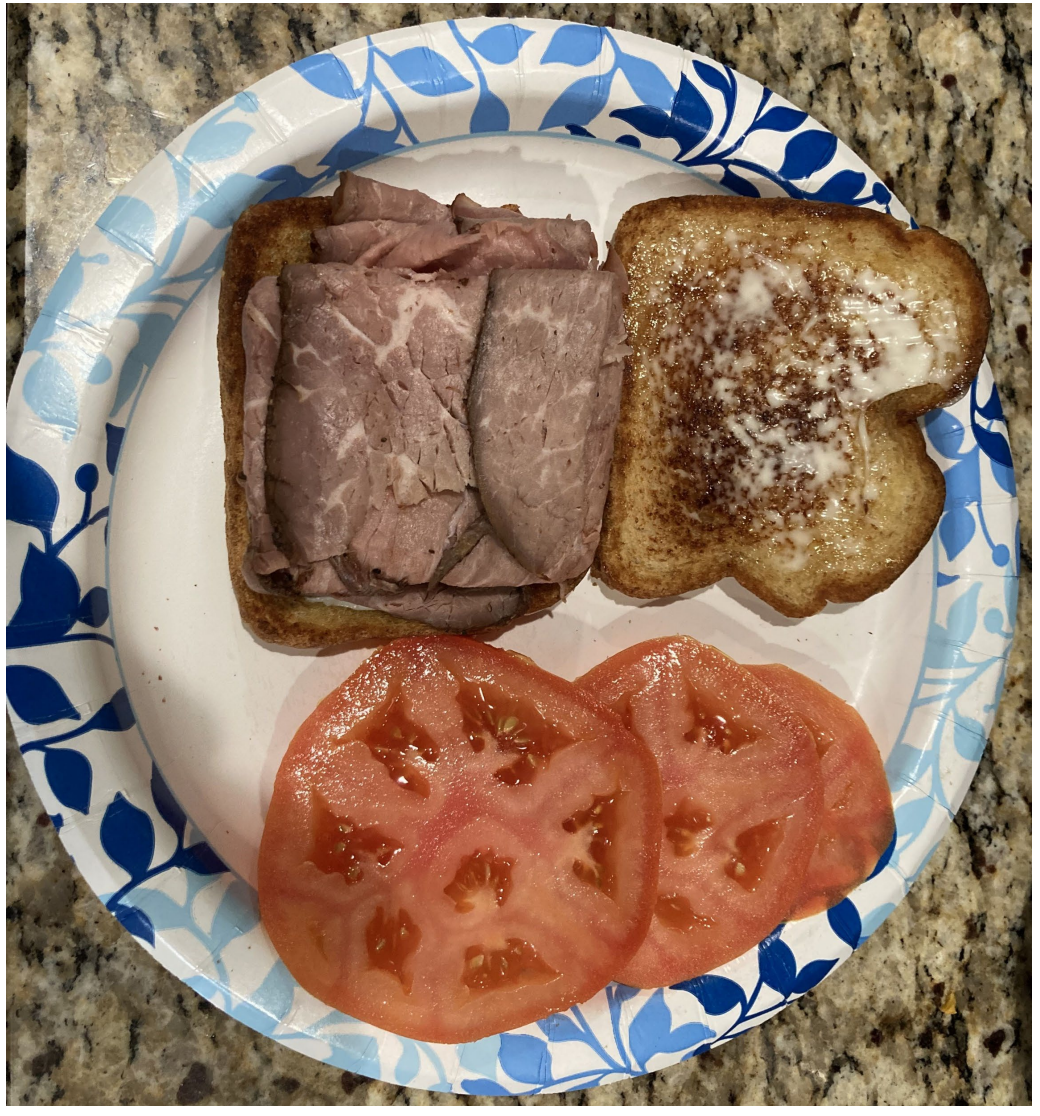


Thank you for attending!!!

Please consider the resources below:

- Today's Download Package
- STM32C071RB Datasheet
- NUCLEO-C071RB Schematic
- WizFi360 User Manual

MORE TO COME..



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

informa markets