CEC Continuing Education Center

DesignNews

Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

Day 5:

The RP2040, Arduino and Radar

Sponsored by

DigiKey

informa markets

Webinar Logistics

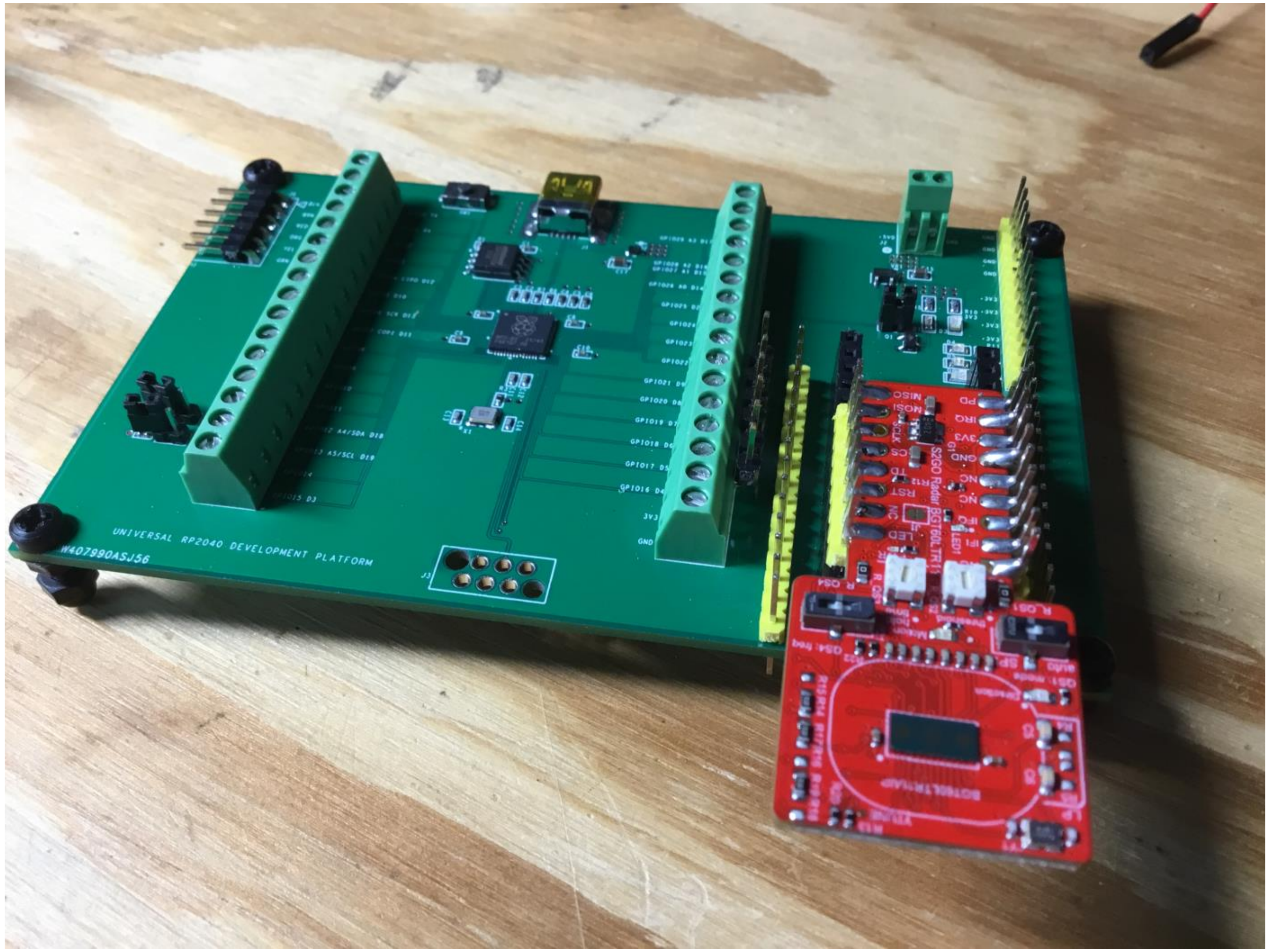
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

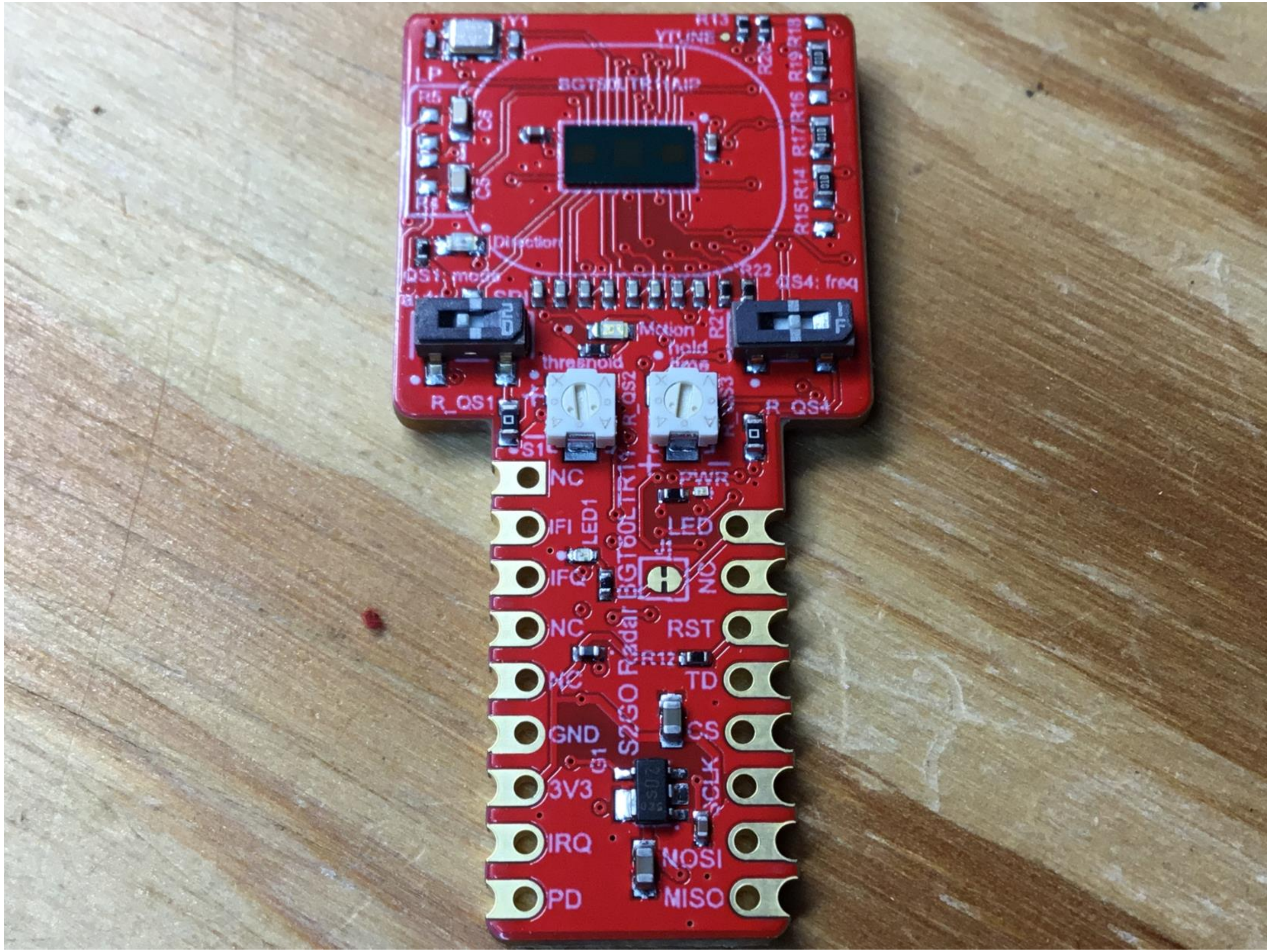
Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

- RP2040 Arduino Radar Hardware
- Arduino Radar Library



RP2040 Arduino Radar Hardware



RP2040 Arduino Radar Hardware

The

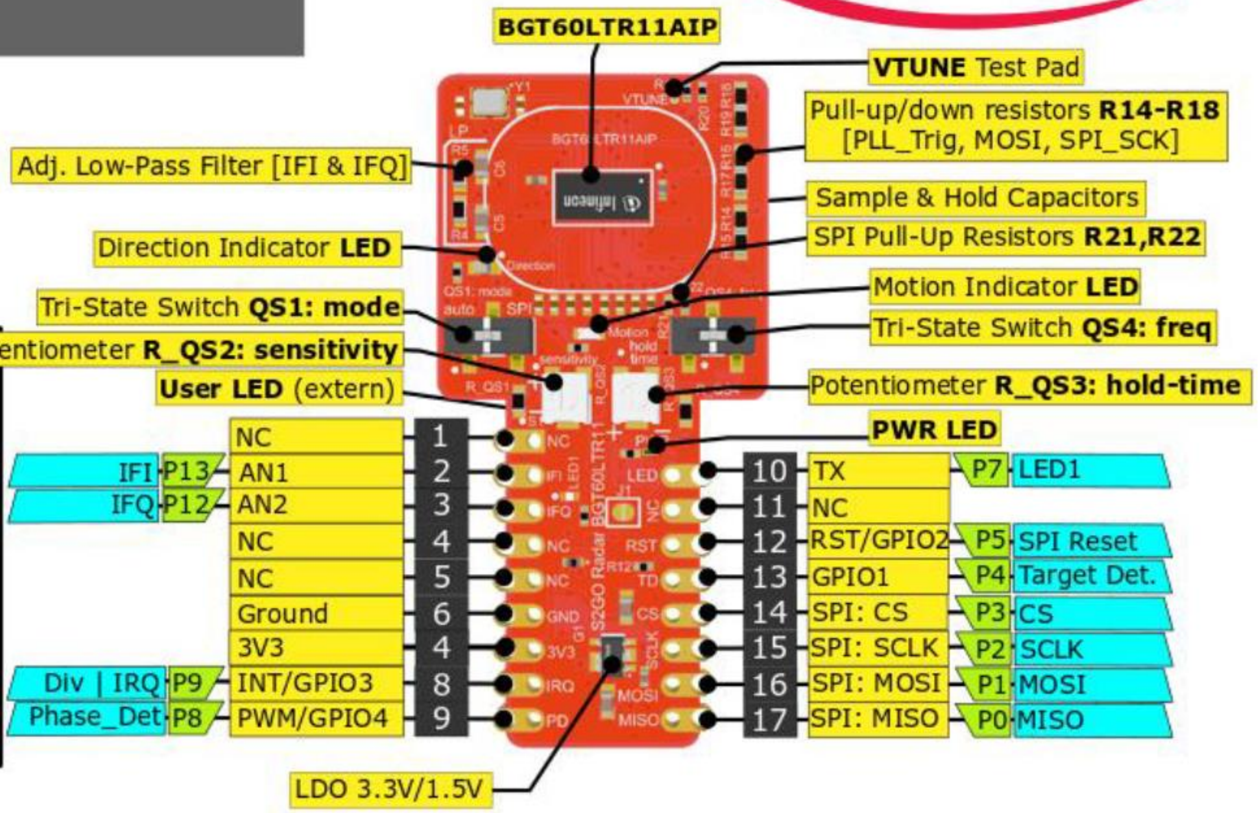
BGT60LTR11AIP Radar Shield2Go

Infineon

ⓘ Ground pins on board connected with each other.

Legend

🟨	Information
▱	Labelling of Pins in Datasheet
▱	Pin Number in Arduino IDE
■	Physical Pin Number
⚠	Warning
ⓘ	Additional Information
NC	Not Connected



BGT60LTR11AIP

VTUNE Test Pad

Pull-up/down resistors **R14-R18** [PLL_Trig, MOSI, SPI_SCK]

Adj. Low-Pass Filter [IFI & IFQ]

Sample & Hold Capacitors

SPI Pull-Up Resistors **R21,R22**

Direction Indicator **LED**

Motion Indicator **LED**

Tri-State Switch **QS1: mode**

Tri-State Switch **QS4: freq**

Potentiometer **R_QS2: sensitivity**

User LED (extern)

Potentiometer **R_QS3: hold-time**

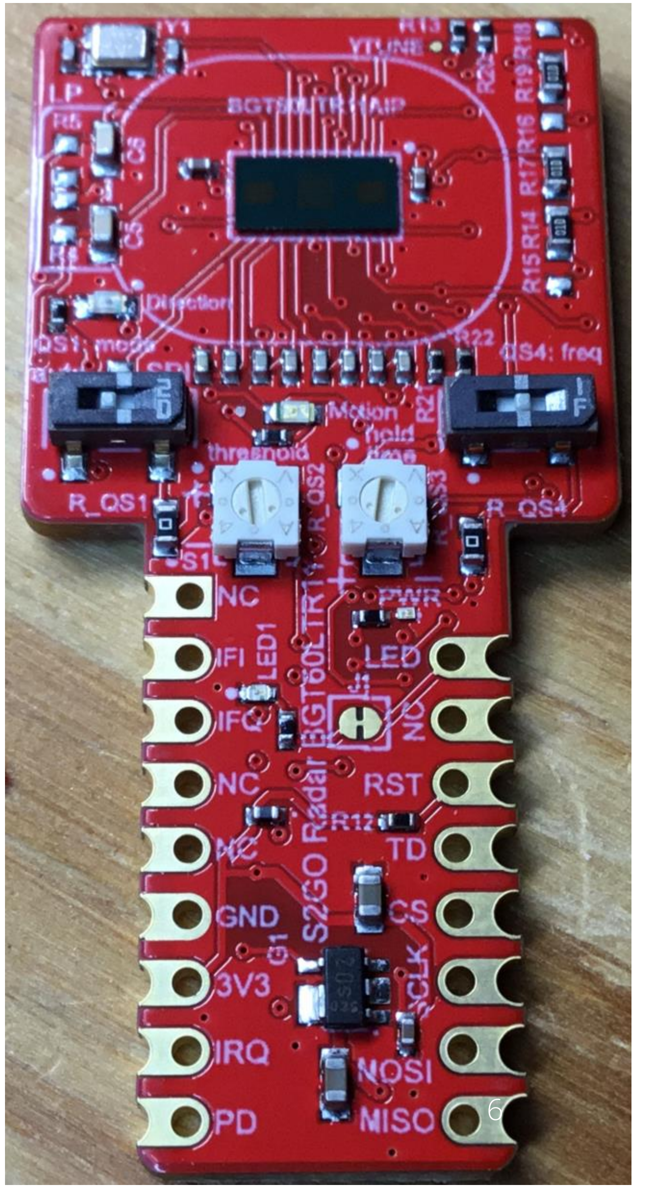
PWR LED

LDO 3.3V/1.5V

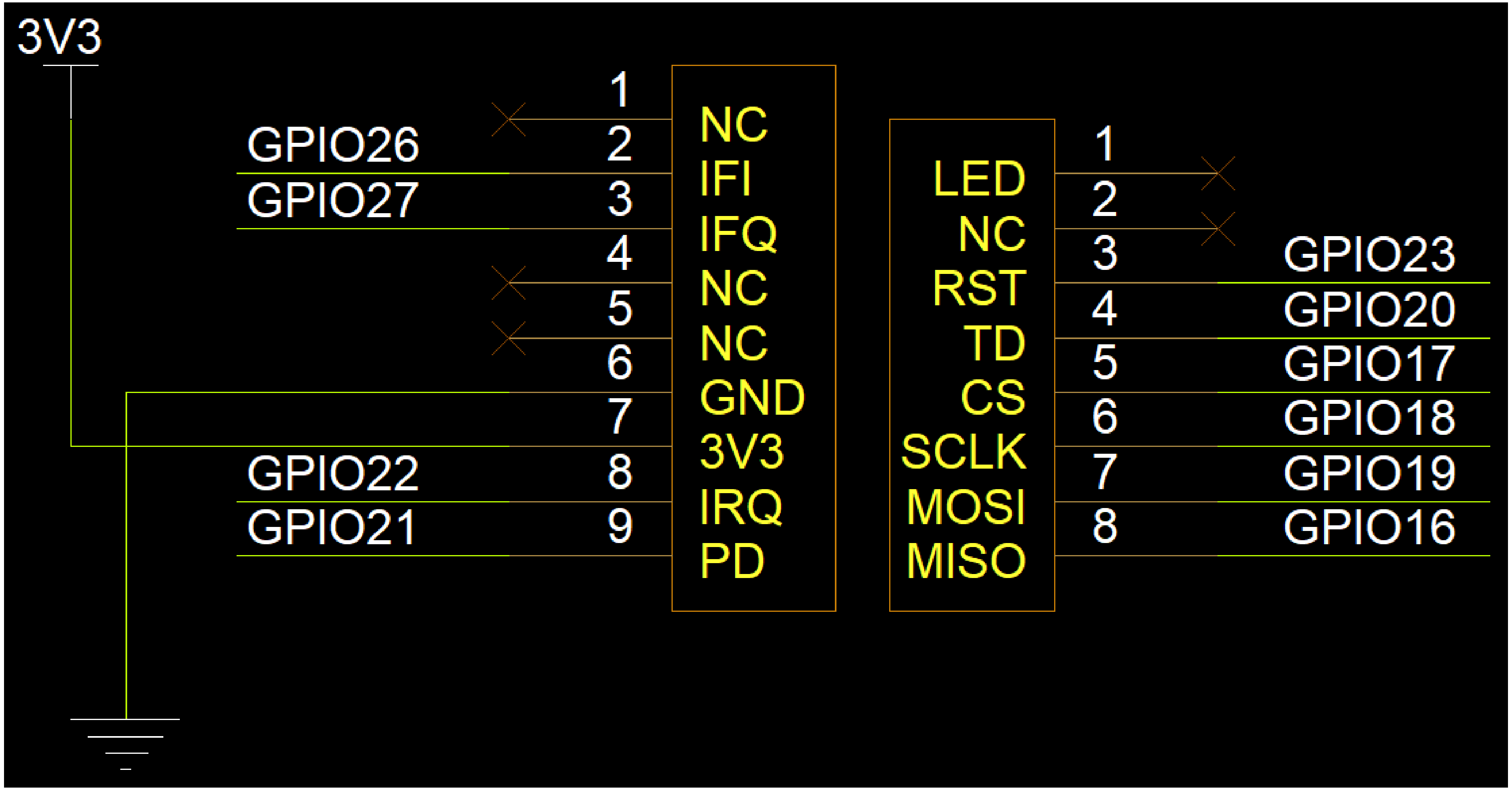
Datasheet	Arduino IDE	Information	Physical Pin
		NC	1
IFI	P13	AN1	2
IFQ	P12	AN2	3
		NC	4
		NC	5
		Ground	6
		3V3	4
Div | IRQ	P9	INT/GPIO3	8
Phase_Det	P8	PWM/GPIO4	9

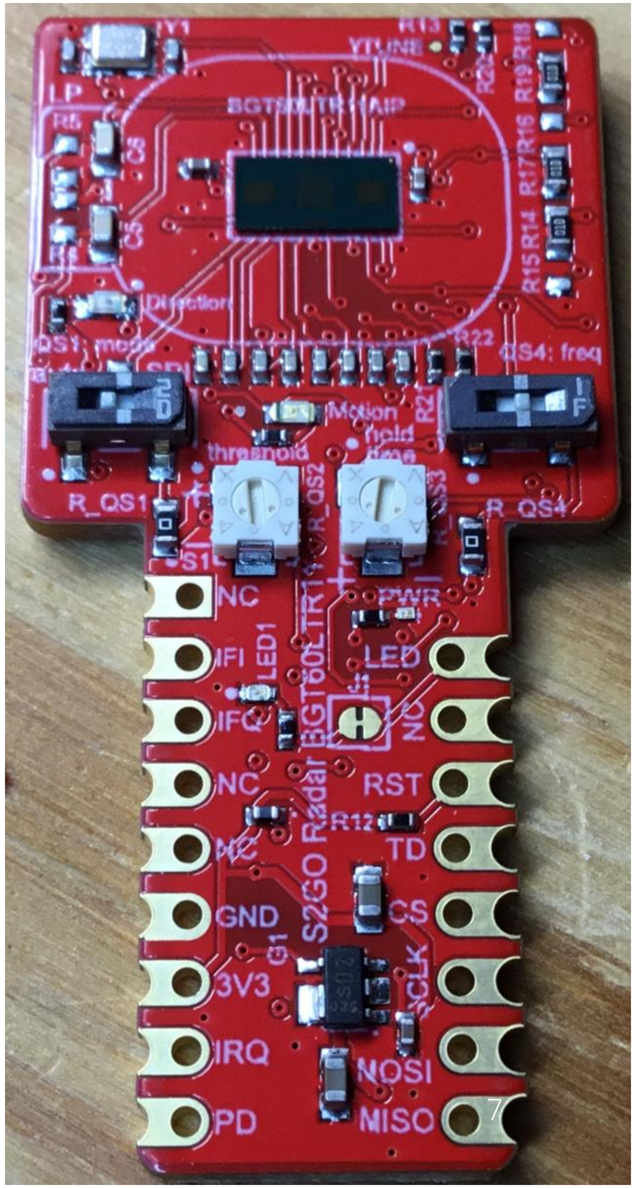
Physical Pin	Information	Arduino IDE	Datasheet
10	TX	P7	LED1
11	NC		
12	RST/GPIO2	P5	SPI Reset
13	GPIO1	P4	Target Det.
14	SPI: CS	P3	CS
15	SPI: SCLK	P2	SCLK
16	SPI: MOSI	P1	MOSI
17	SPI: MISO	P0	MISO

⚠ The maximum voltage on any pin is 4 V.

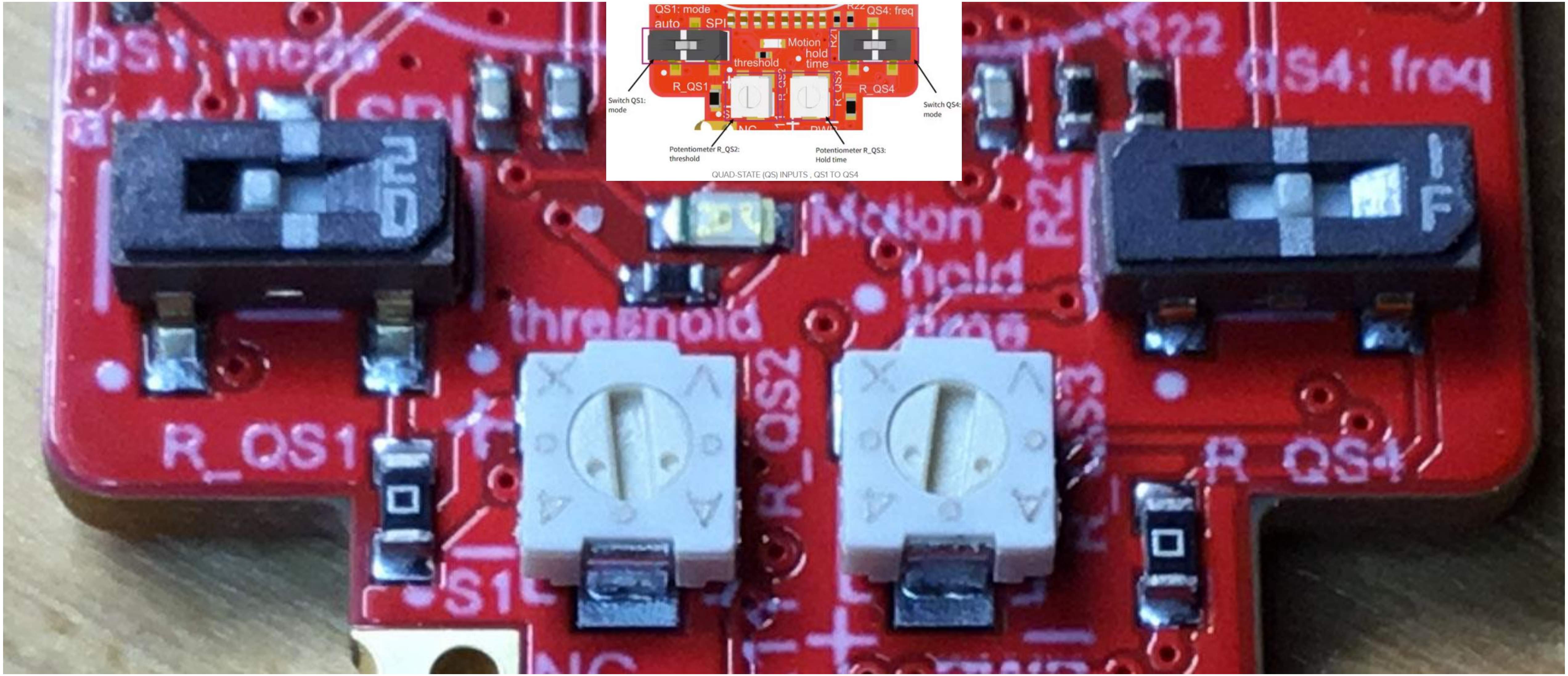


RP2040 Arduino Radar Hardware



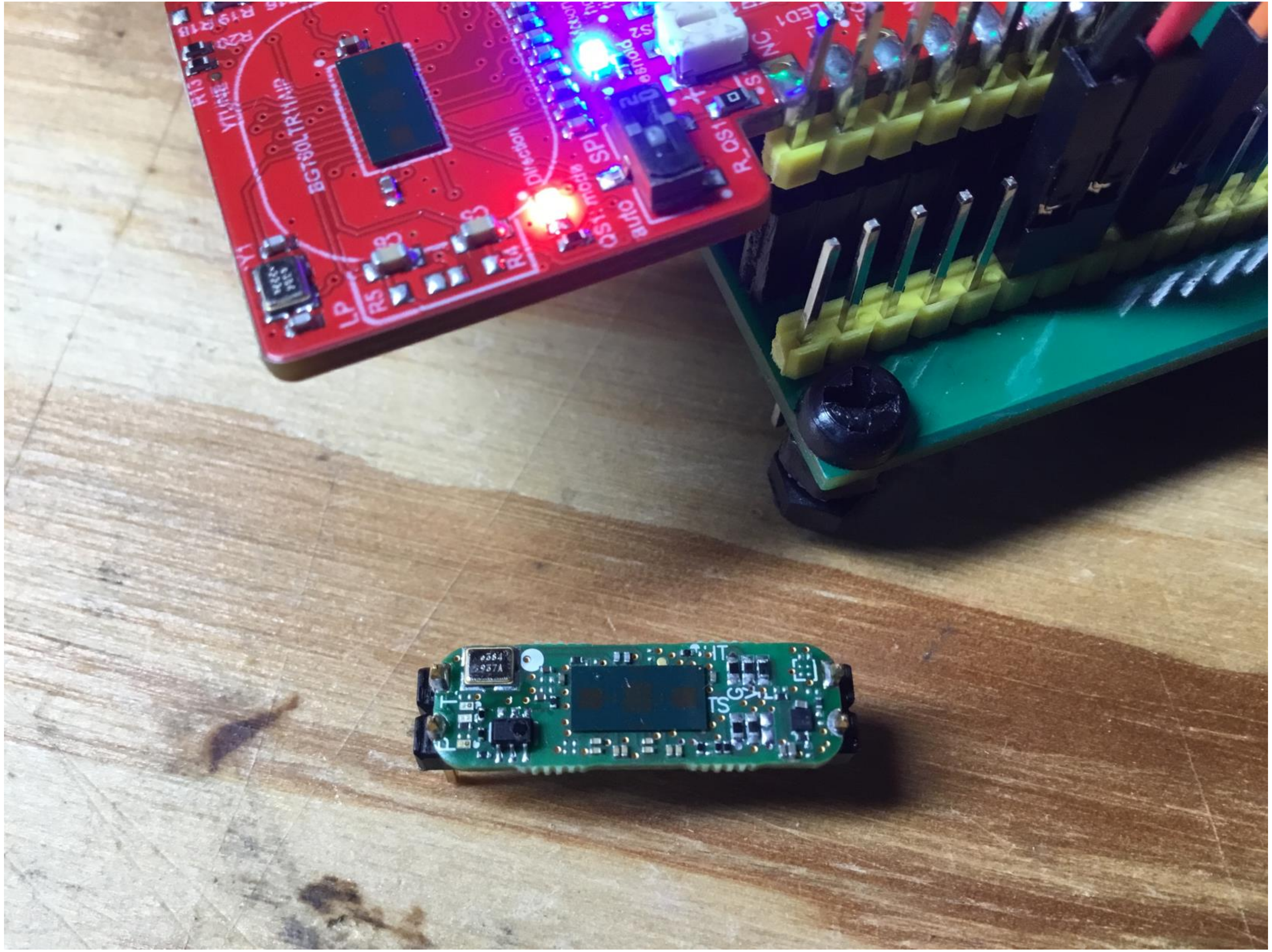


RP2040 Arduino Radar Hardware

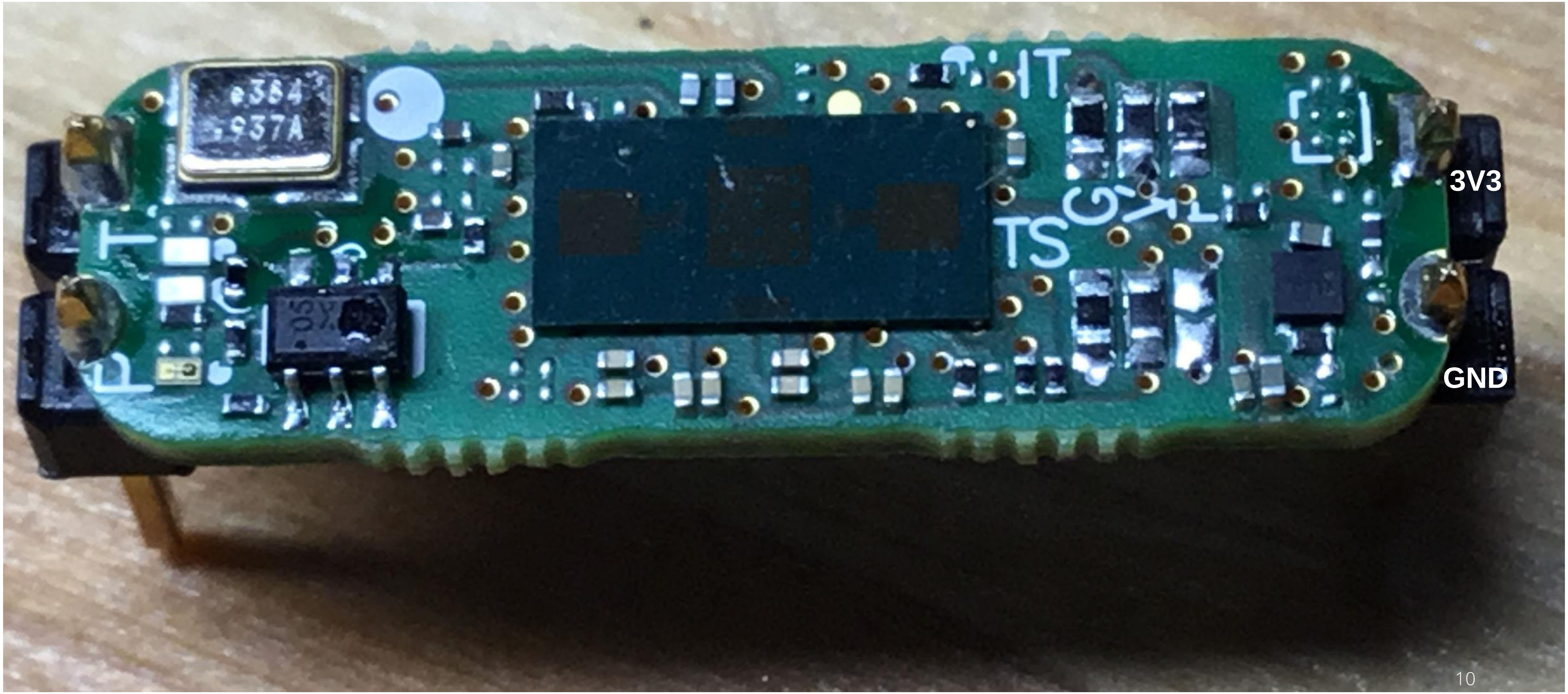


QUAD-STATE (QS) INPUTS , QS1 TO QS4

RP2040 Arduino Radar Hardware



RP2040 Arduino Radar Hardware



RP2040 Arduino Radar Hardware

TD = Target Detection

TD: Logic HIGH when there has been no Target/Motion detected and logic LOW when there has been a Target/Motion detected.

PD = Phase Detection

PD: If motion has been detected: logic HIGH when a target is approaching the Radar and logic LOW when a target is departing.

CEC Continuing Education Center

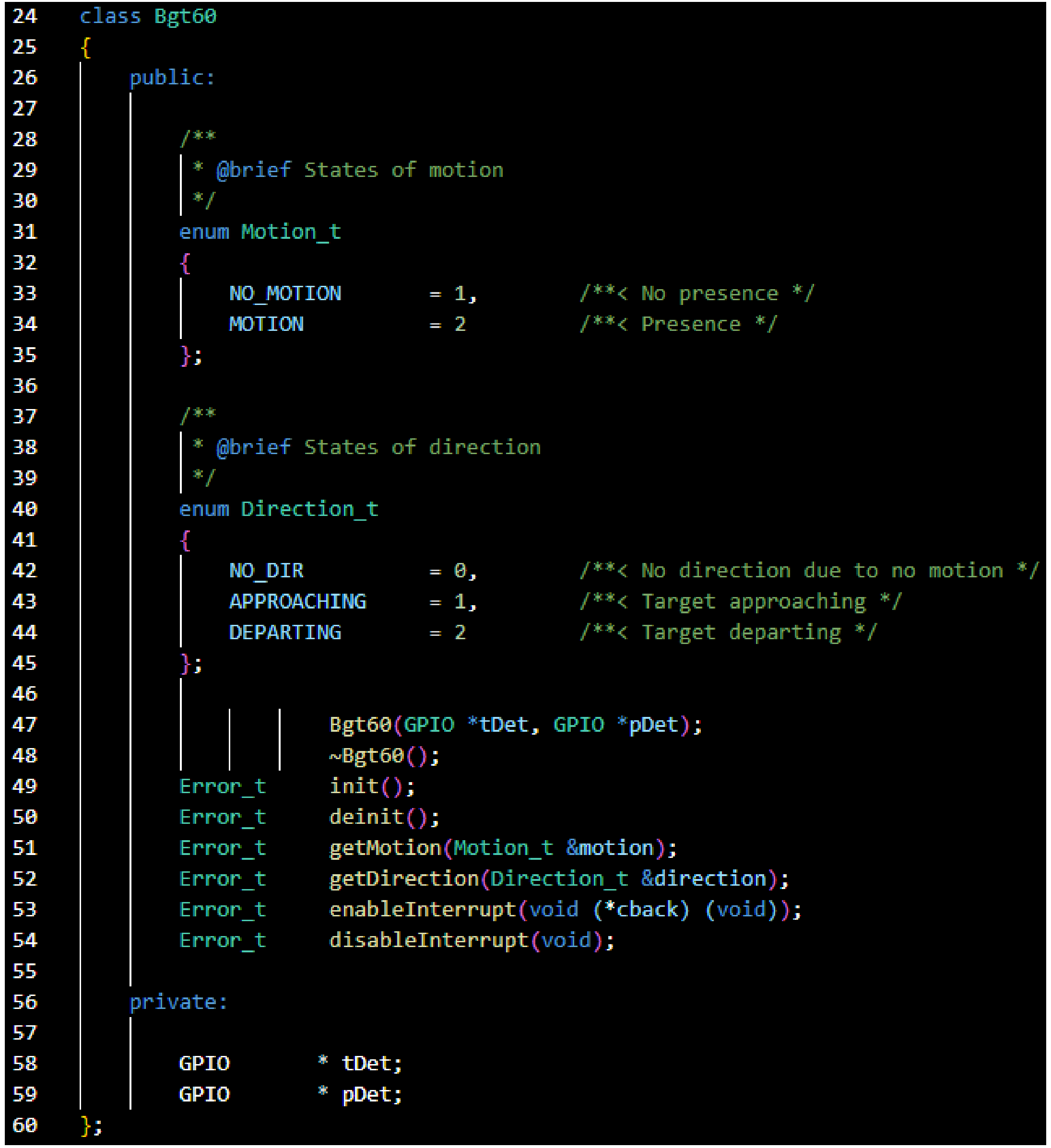
Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

The RP2040, Arduino and Radar

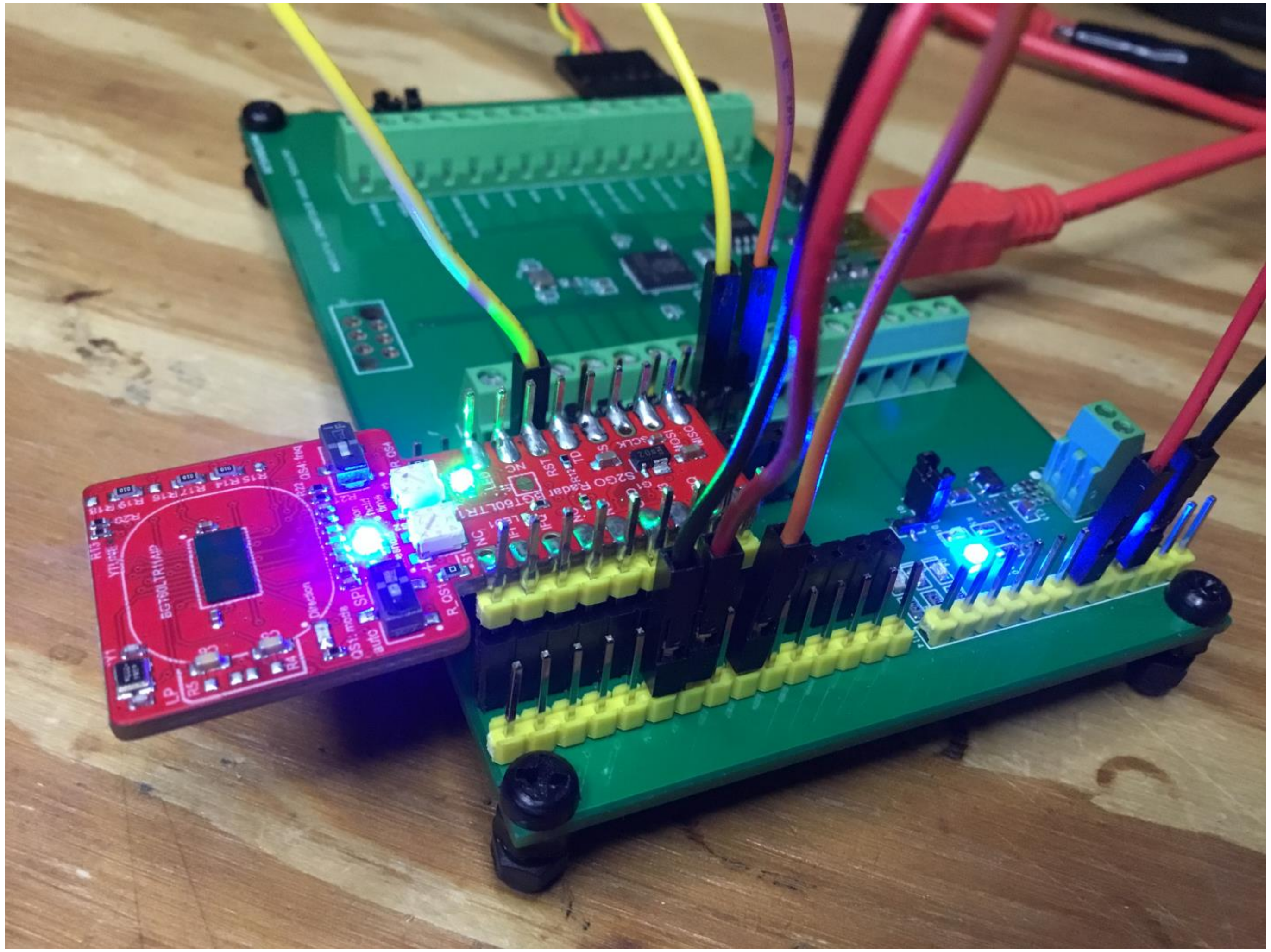
Arduino Radar Library

DigiKey

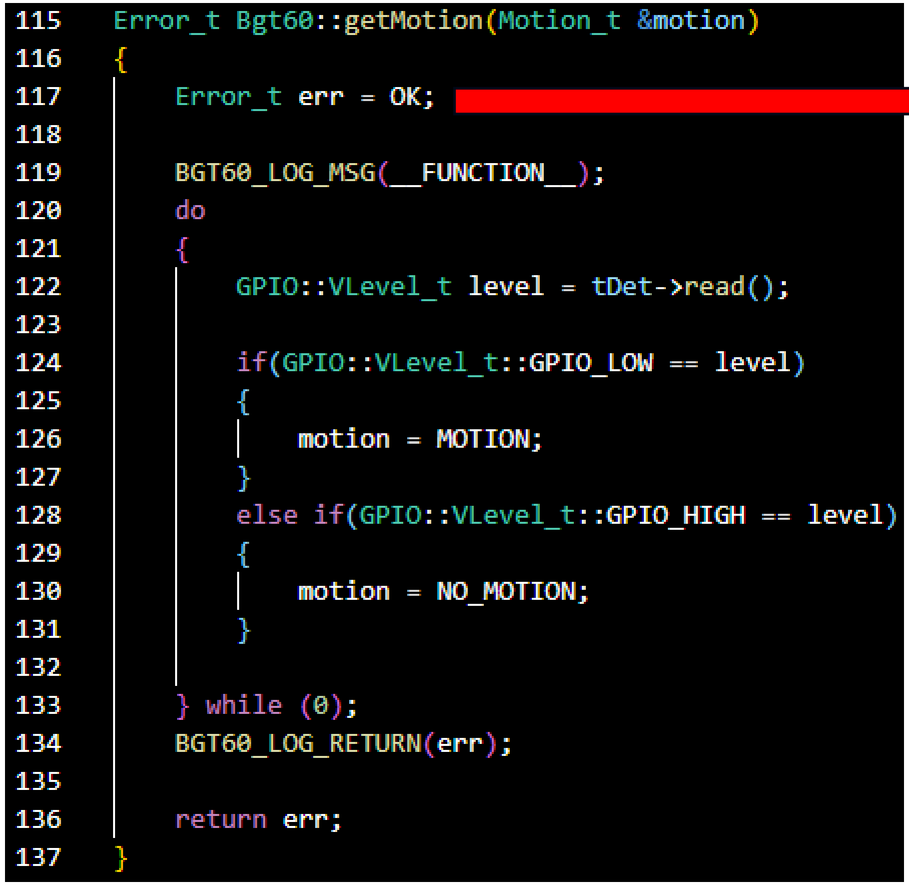
bgt60.hpp



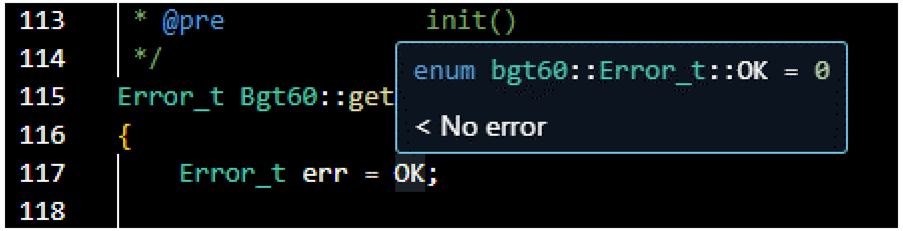
```cpp
class Bgt60
{
    public:

        /**
         * @brief States of motion
         */
        enum Motion_t
        {
            NO_MOTION       = 1,        /**< No presence */
            MOTION          = 2         /**< Presence */
        };

        /**
         * @brief States of direction
         */
        enum Direction_t
        {
            NO_DIR          = 0,        /**< No direction due to no motion */
            APPROACHING     = 1,        /**< Target approaching */
            DEPARTING       = 2         /**< Target departing */
        };

                    Bgt60(GPIO *tDet, GPIO *pDet);
                    ~Bgt60();
        Error_t     init();
        Error_t     deinit();
        Error_t     getMotion(Motion_t &motion);
        Error_t     getDirection(Direction_t &direction);
        Error_t     enableInterrupt(void (*cback) (void));
        Error_t     disableInterrupt(void);

    private:

        GPIO       * tDet;
        GPIO       * pDet;
};
```



bgt60.cpp - getMotion

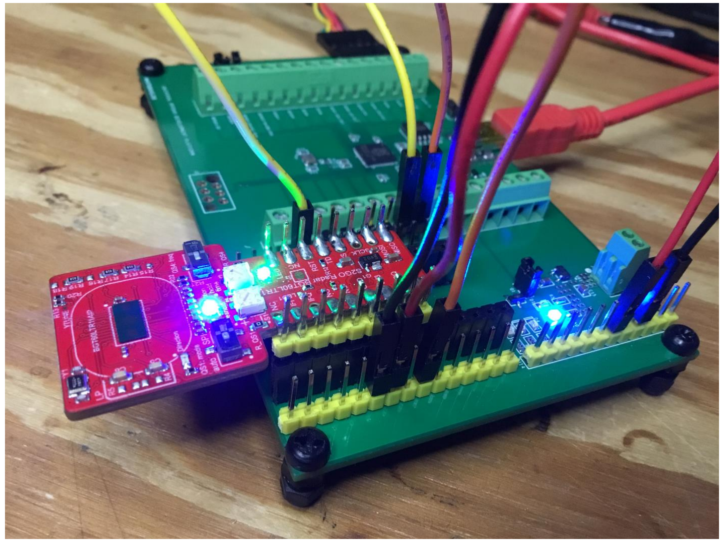


```
Error_t Bgt60::getMotion(Motion_t &motion)
{
    Error_t err = OK;

    BGT60_LOG_MSG(__FUNCTION__);
    do
    {
        GPIO::VLevel_t level = tDet->read();

        if(GPIO::VLevel_t::GPIO_LOW == level)
        {
            motion = MOTION;
        }
        else if(GPIO::VLevel_t::GPIO_HIGH == level)
        {
            motion = NO_MOTION;
        }

    } while (0);
    BGT60_LOG_RETURN(err);

    return err;
}
```

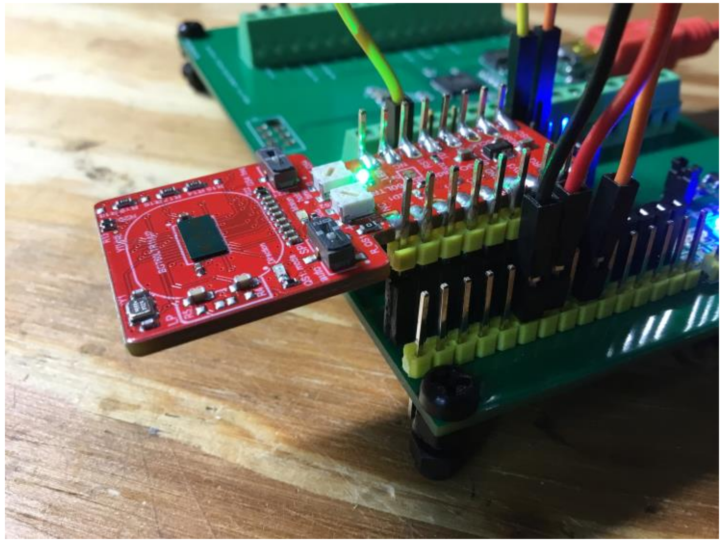


```
 * @pre            init()
 */
Error_t Bgt60::get
{
    Error_t err = OK;

```

```
enum bgt60::Error_t::OK = 0
< No error
```

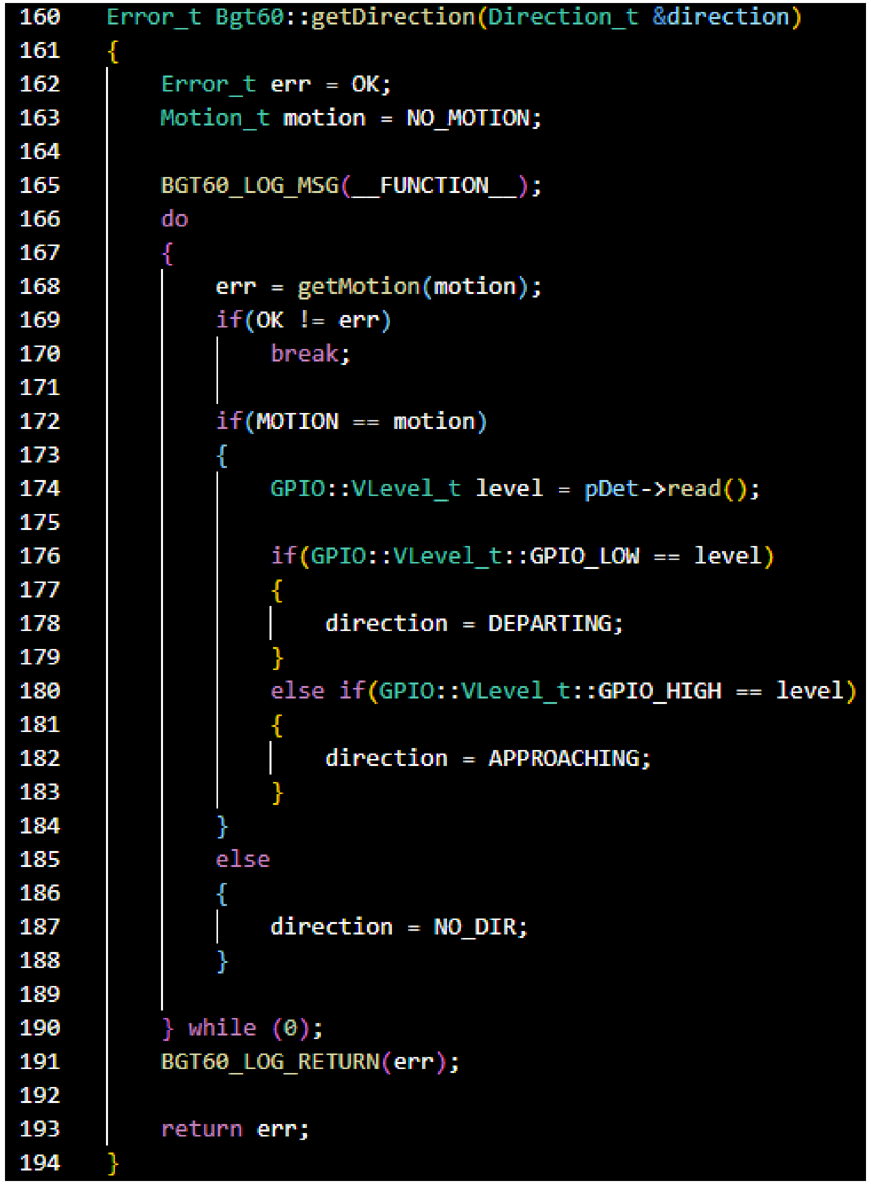


MOTION



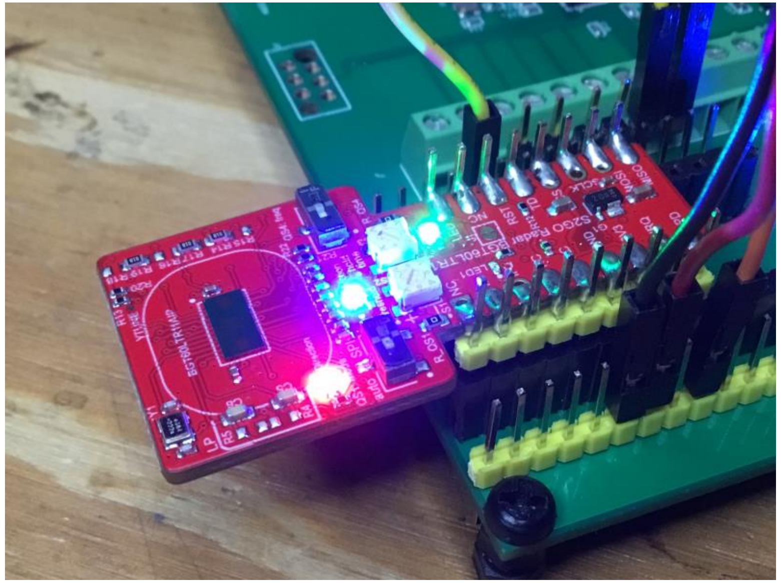
NO MOTION

bgt60.cpp - getDirection

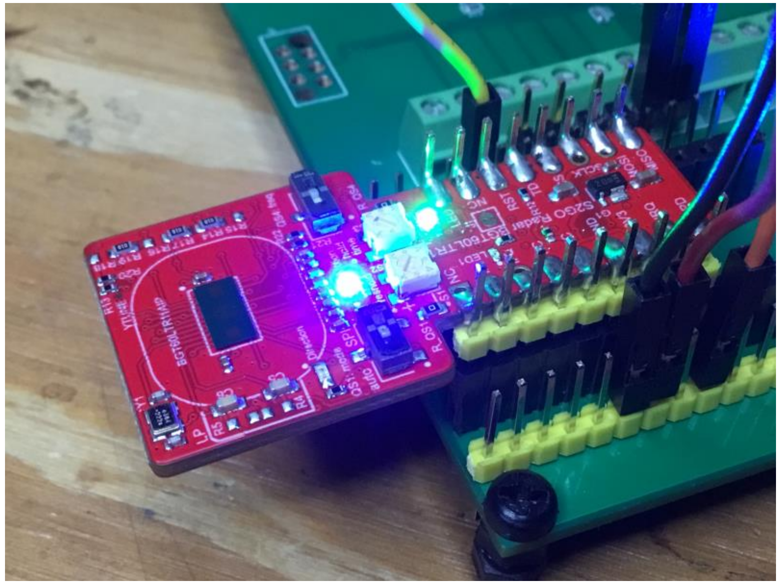


```
Error_t Bgt60::getDirection(Direction_t &direction)
{
    Error_t err = OK;
    Motion_t motion = NO_MOTION;

    BGT60_LOG_MSG(__FUNCTION__);
    do
    {
        err = getMotion(motion);
        if(OK != err)
            break;

        if(MOTION == motion)
        {
            GPIO::VLevel_t level = pDet->read();

            if(GPIO::VLevel_t::GPIO_LOW == level)
            {
                direction = DEPARTING;
            }
            else if(GPIO::VLevel_t::GPIO_HIGH == level)
            {
                direction = APPROACHING;
            }
        }
        else
        {
            direction = NO_DIR;
        }

    } while (0);
    BGT60_LOG_RETURN(err);

    return err;
}
```

DEPARTING



APPROACHING

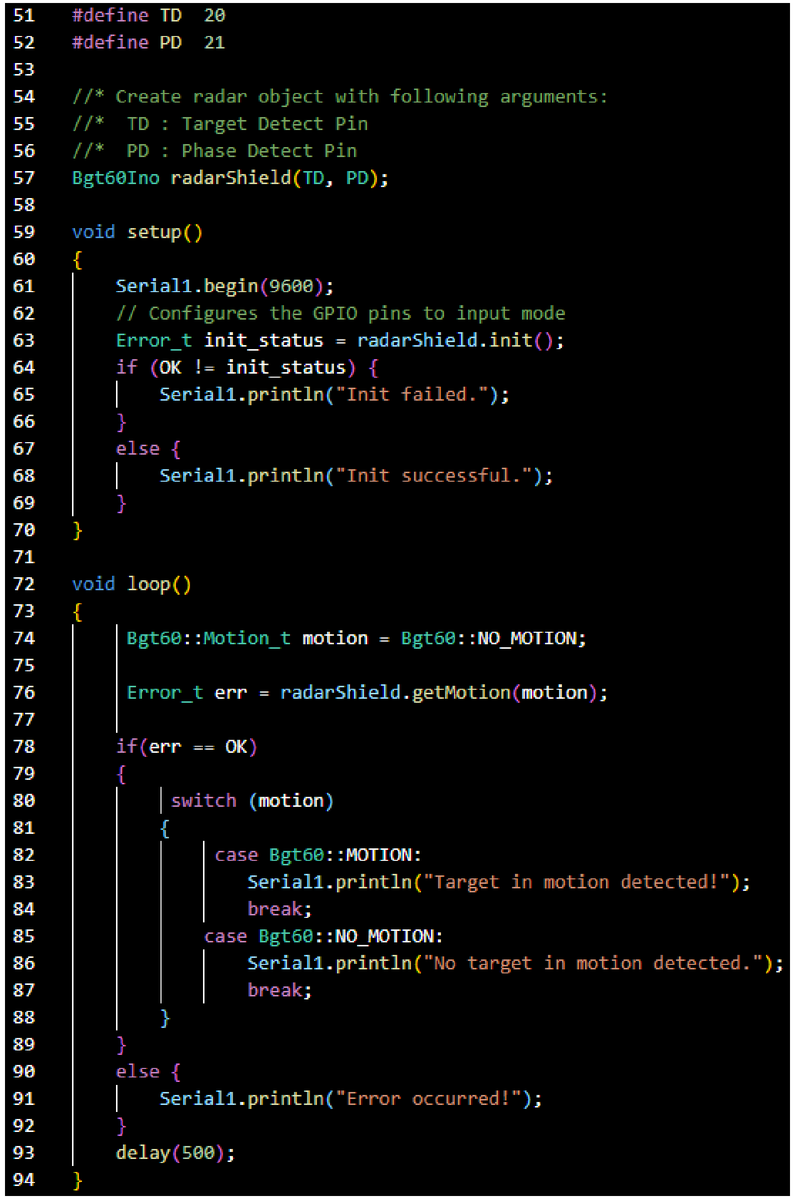


Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

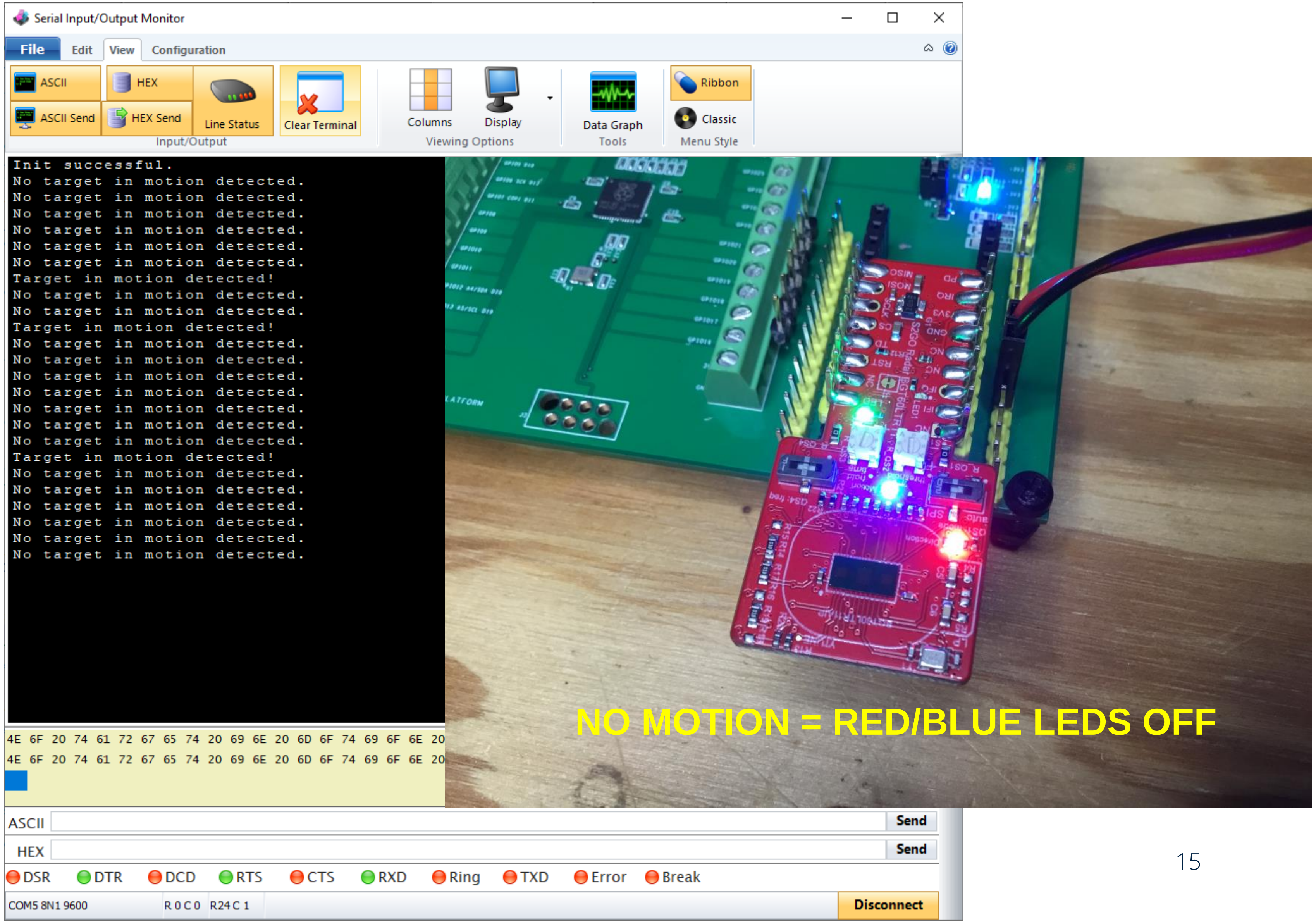
The RP2040, Arduino and Radar

Arduino Radar Library

Motion Detection



```
#define TD  20
#define PD  21

//* Create radar object with following arguments:
//*   TD : Target Detect Pin
//*   PD : Phase Detect Pin
Bgt60Ino radarShield(TD, PD);

void setup()
{
    Serial1.begin(9600);
    // Configures the GPIO pins to input mode
    Error_t init_status = radarShield.init();
    if (OK != init_status) {
        Serial1.println("Init failed.");
    }
    else {
        Serial1.println("Init successful.");
    }
}

void loop()
{
     Bgt60::Motion_t motion = Bgt60::NO_MOTION;

     Error_t err = radarShield.getMotion(motion);

    if(err == OK)
    {
        switch (motion)
        {
            case Bgt60::MOTION:
                Serial1.println("Target in motion detected!");
                break;
           case Bgt60::NO_MOTION:
                Serial1.println("No target in motion detected.");
                break;
        }
    }
    else {
        Serial1.println("Error occurred!");
    }
    delay(500);
}
```



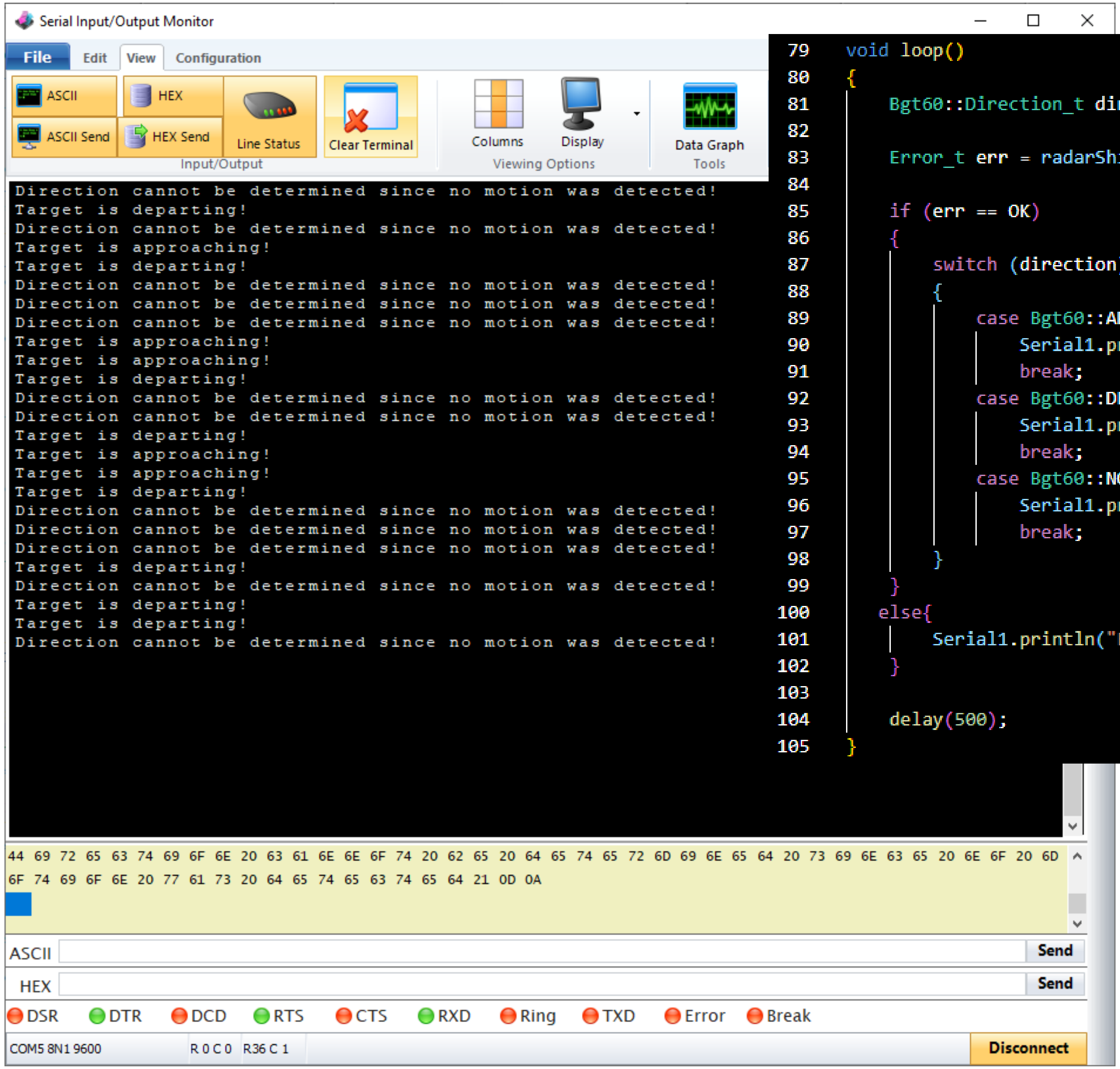
```
Init successful.
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
Target in motion detected!
No target in motion detected.
No target in motion detected.
Target in motion detected!
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
Target in motion detected!
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
No target in motion detected.
```

```
4E 6F 20 74 61 72 67 65 74 20 69 6E 20 6D 6F 74 69 6F 6E 20
4E 6F 20 74 61 72 67 65 74 20 69 6E 20 6D 6F 74 69 6F 6E 20
```

COM5 8N1 9600   R 0 C 0   R24 C 1

NO MOTION = RED/BLUE LEDS OFF

Direction Detection

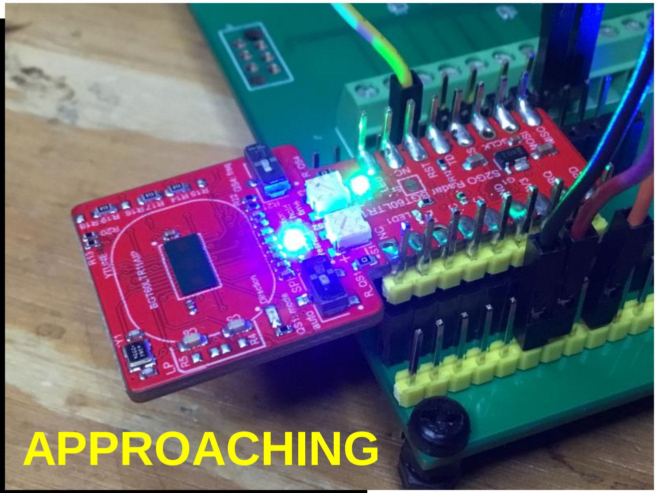


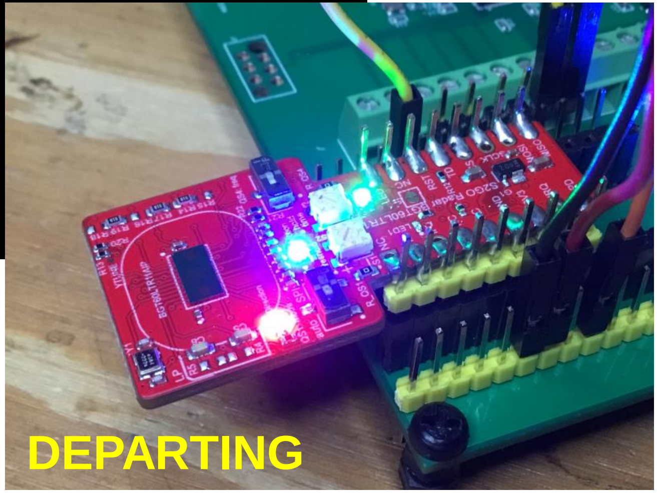
```
Direction cannot be determined since no motion was detected!
Target is departing!
Direction cannot be determined since no motion was detected!
Target is approaching!
Target is departing!
Direction cannot be determined since no motion was detected!
Direction cannot be determined since no motion was detected!
Direction cannot be determined since no motion was detected!
Target is approaching!
Target is approaching!
Target is departing!
Direction cannot be determined since no motion was detected!
Direction cannot be determined since no motion was detected!
Target is departing!
Target is approaching!
Target is approaching!
Target is departing!
Direction cannot be determined since no motion was detected!
Direction cannot be determined since no motion was detected!
Direction cannot be determined since no motion was detected!
Target is departing!
Direction cannot be determined since no motion was detected!
Target is departing!
Target is departing!
Direction cannot be determined since no motion was detected!
```

```
44 69 72 65 63 74 69 6F 6E 20 63 61 6E 6E 6F 74 20 62 65 20 64 65 74 65 72 6D 69 6E 65 64 20 73 69 6E 63 65 20 6E 6F 20 6D
6F 74 69 6F 6E 20 77 61 73 20 64 65 74 65 63 74 65 64 21 0D 0A
```

COM5 8N1 9600 R 0 C 0 R36 C 1

```
void loop()
{
    Bgt60::Direction_t direction = Bgt60::NO_DIR;

    Error_t err = radarShield.getDirection(direction);

    if (err == OK)
    {
        switch (direction)
        {
            case Bgt60::APPROACHING:
                Serial1.println("Target is approaching!");
                break;
            case Bgt60::DEPARTING:
                Serial1.println("Target is departing!");
                break;
            case Bgt60::NO_DIR:
                Serial1.println("Direction cannot be determined since no motion was detected!");
                break;
        }
    }
   else{
        Serial1.println("Error occurred!");
    }

    delay(500);
}
```



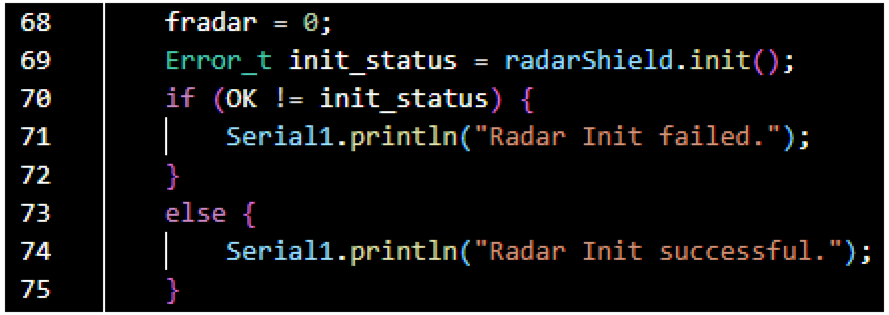


Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

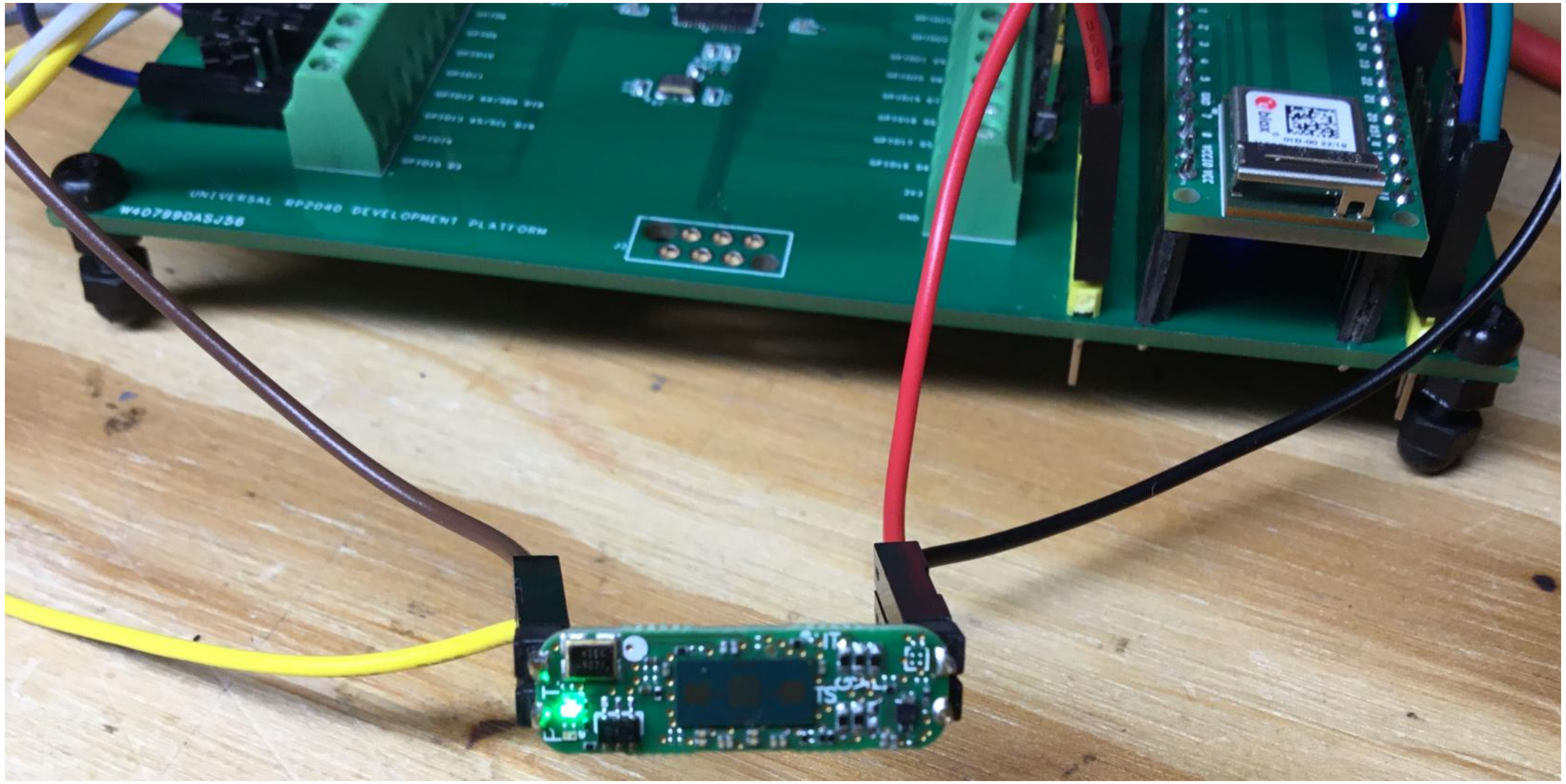
The RP2040, Arduino and Radar

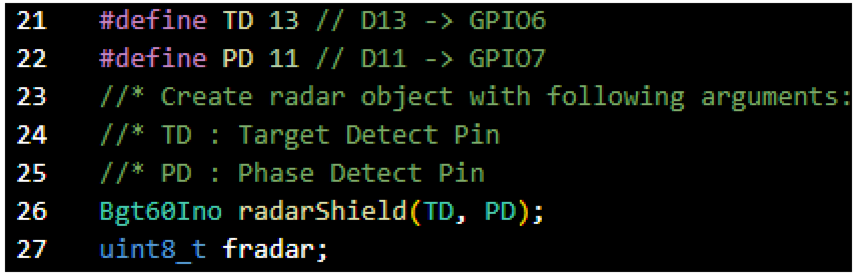
Arduino Radar Library

Radar Over UDP

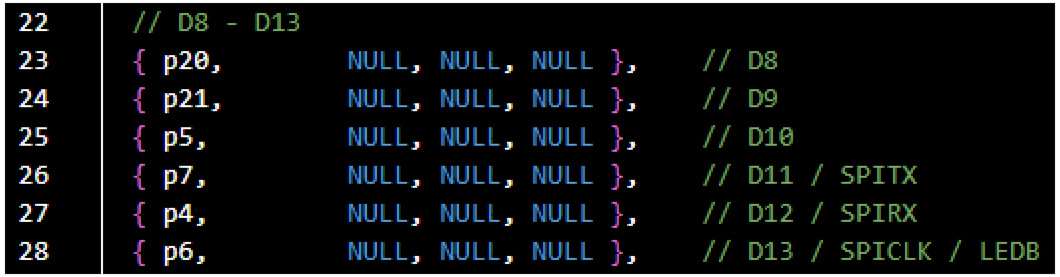


```
    fradar = 0;
    Error_t init_status = radarShield.init();
    if (OK != init_status) {
        Serial1.println("Radar Init failed.");
    }
    else {
        Serial1.println("Radar Init successful.");
    }
```



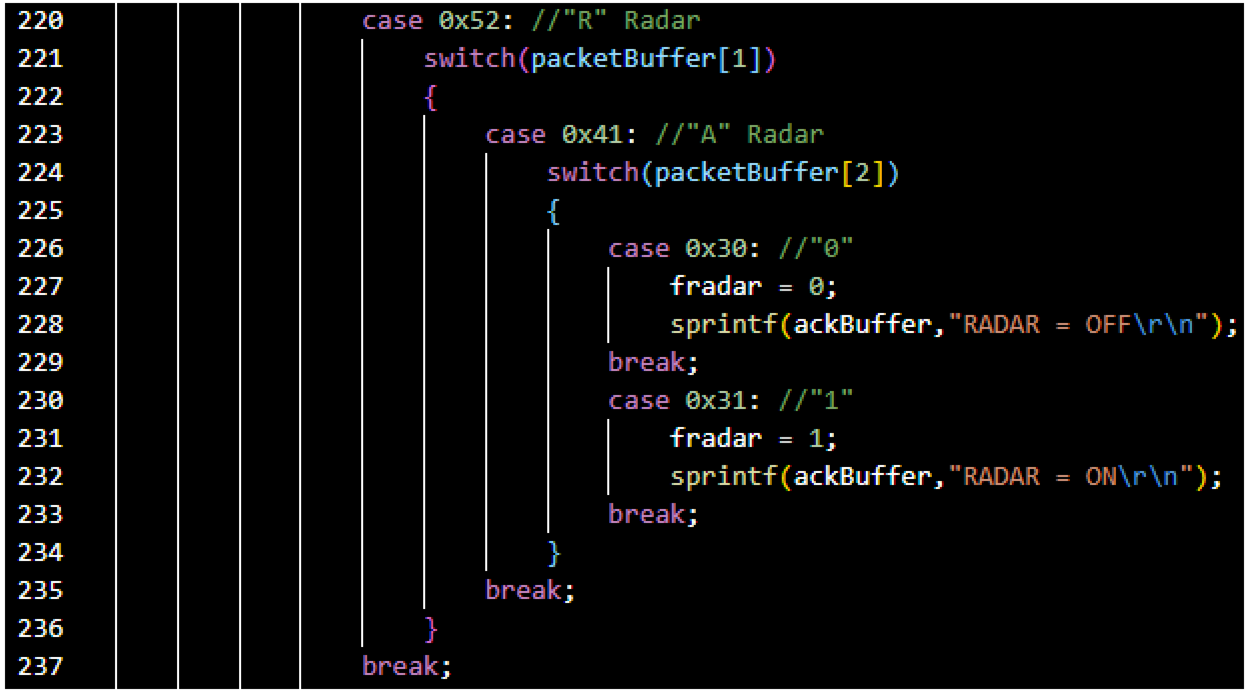


```
#define TD 13 // D13 -> GPIO6
#define PD 11 // D11 -> GPIO7
//* Create radar object with following arguments:
//* TD : Target Detect Pin
//* PD : Phase Detect Pin
Bgt60Ino radarShield(TD, PD);
uint8_t fradar;
```

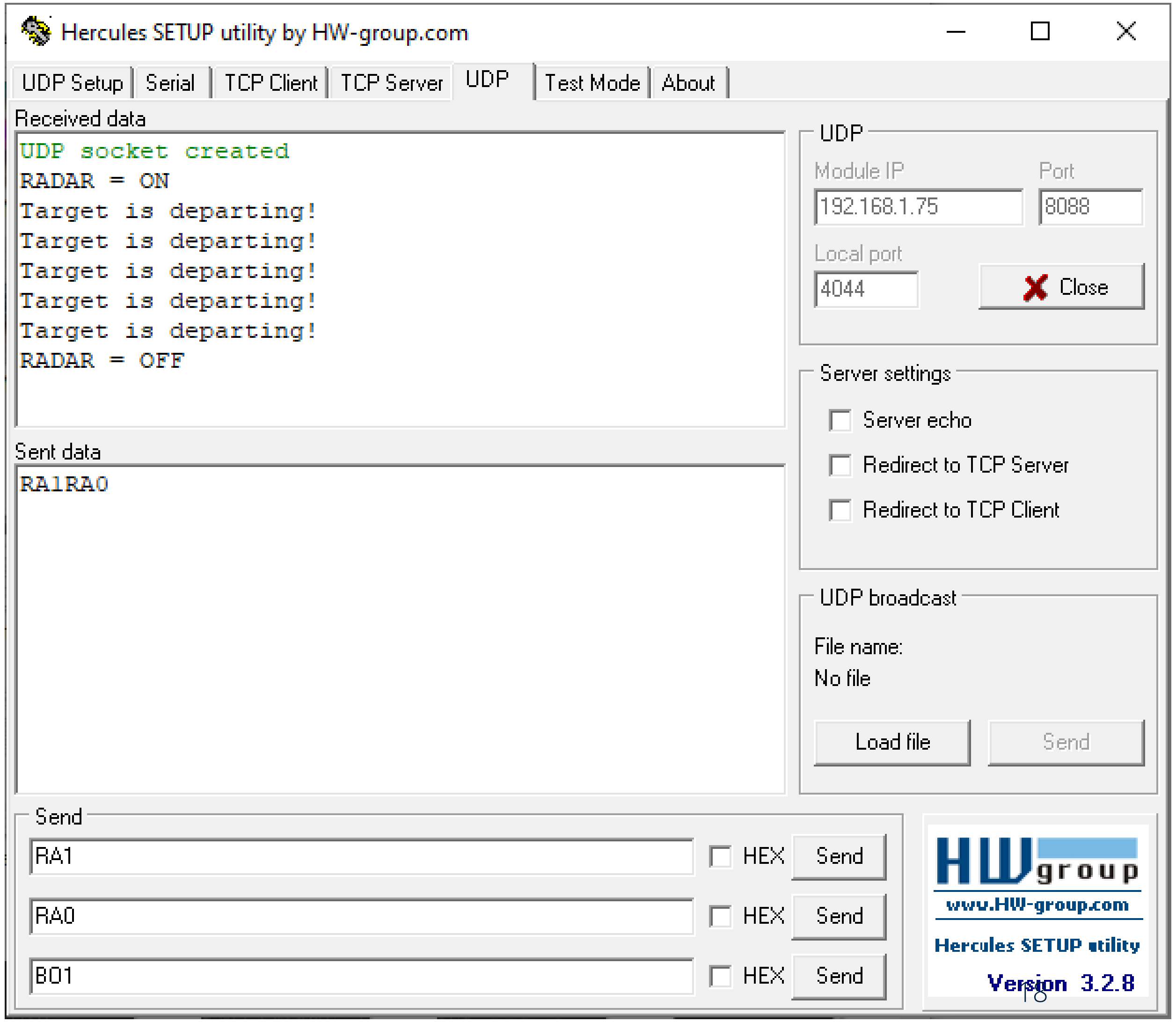


```
  // D8 - D13
  { p20,        NULL, NULL, NULL },    // D8
  { p21,        NULL, NULL, NULL },    // D9
  { p5,         NULL, NULL, NULL },    // D10
  { p7,         NULL, NULL, NULL },    // D11 / SPITX
  { p4,         NULL, NULL, NULL },    // D12 / SPIRX
  { p6,         NULL, NULL, NULL },    // D13 / SPICLK / LEDB
```

Radar Over UDP



```
case 0x52: //"R" Radar
    switch(packetBuffer[1])
    {
        case 0x41: //"A" Radar
            switch(packetBuffer[2])
            {
                case 0x30: //"0"
                    fradar = 0;
                    sprintf(ackBuffer,"RADAR = OFF\r\n");
                break;
                case 0x31: //"1"
                    fradar = 1;
                    sprintf(ackBuffer,"RADAR = ON\r\n");
                break;
            }
        break;
    }
break;
```



Received data

```
UDP socket created
RADAR = ON
Target is departing!
Target is departing!
Target is departing!
Target is departing!
Target is departing!
RADAR = OFF
```

Sent data

```
RA1RA0
```

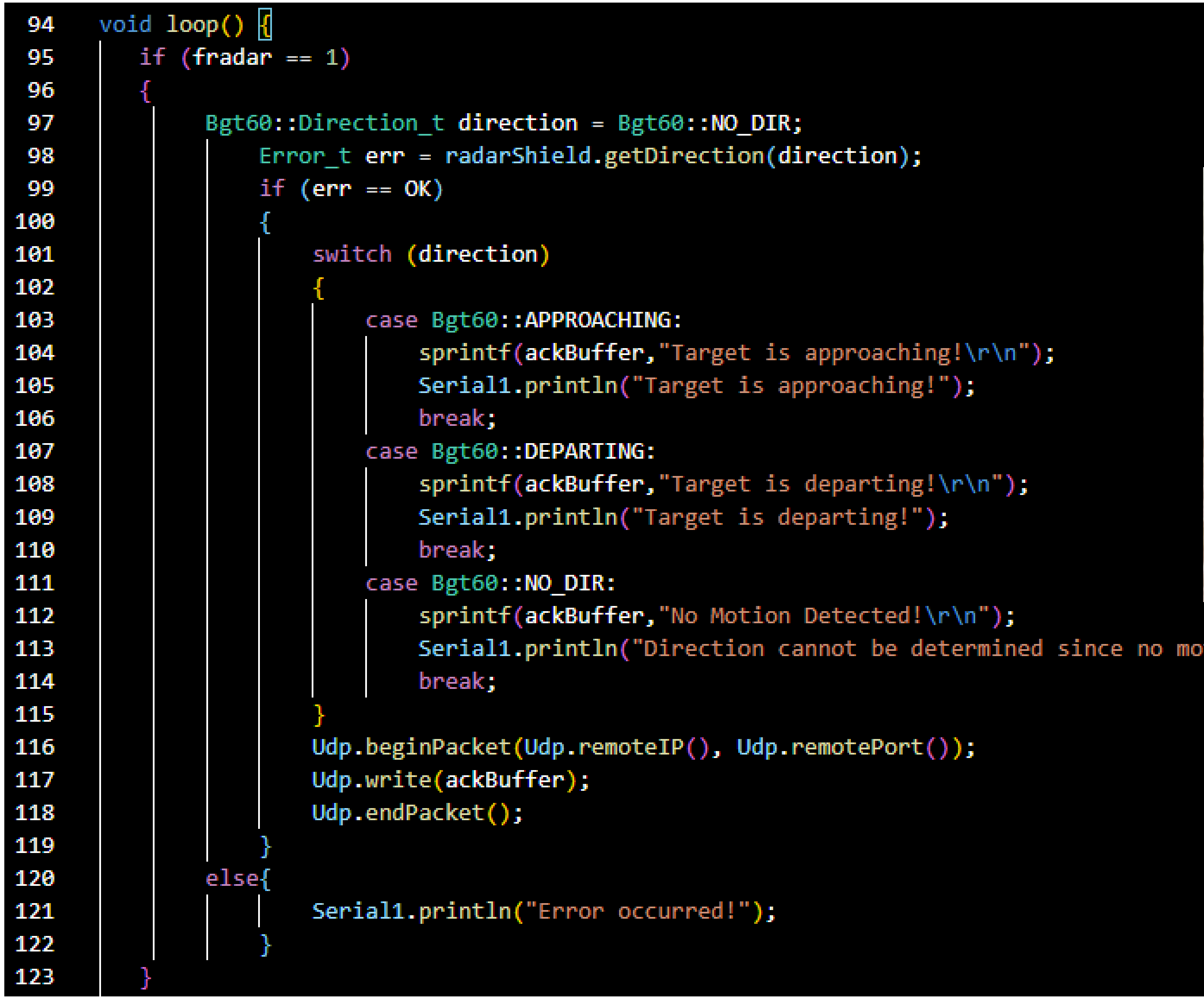
UDP
Module IP: 192.168.1.75
Port: 8088
Local port: 4044

Server settings
□ Server echo
□ Redirect to TCP Server
□ Redirect to TCP Client

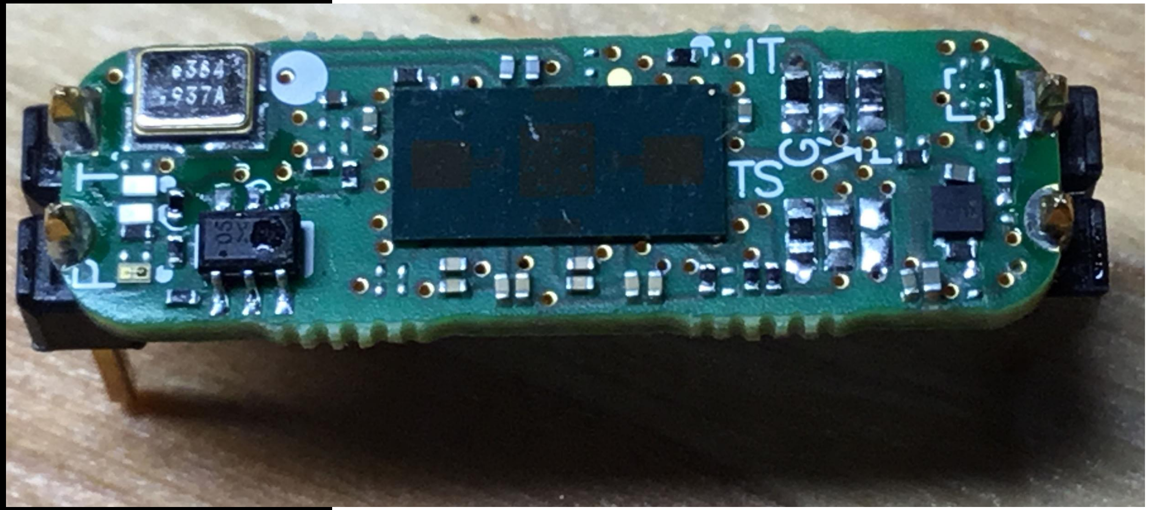
UDP broadcast
File name:
No file

Send
RA1 □ HEX Send
RA0 □ HEX Send
B01 □ HEX Send

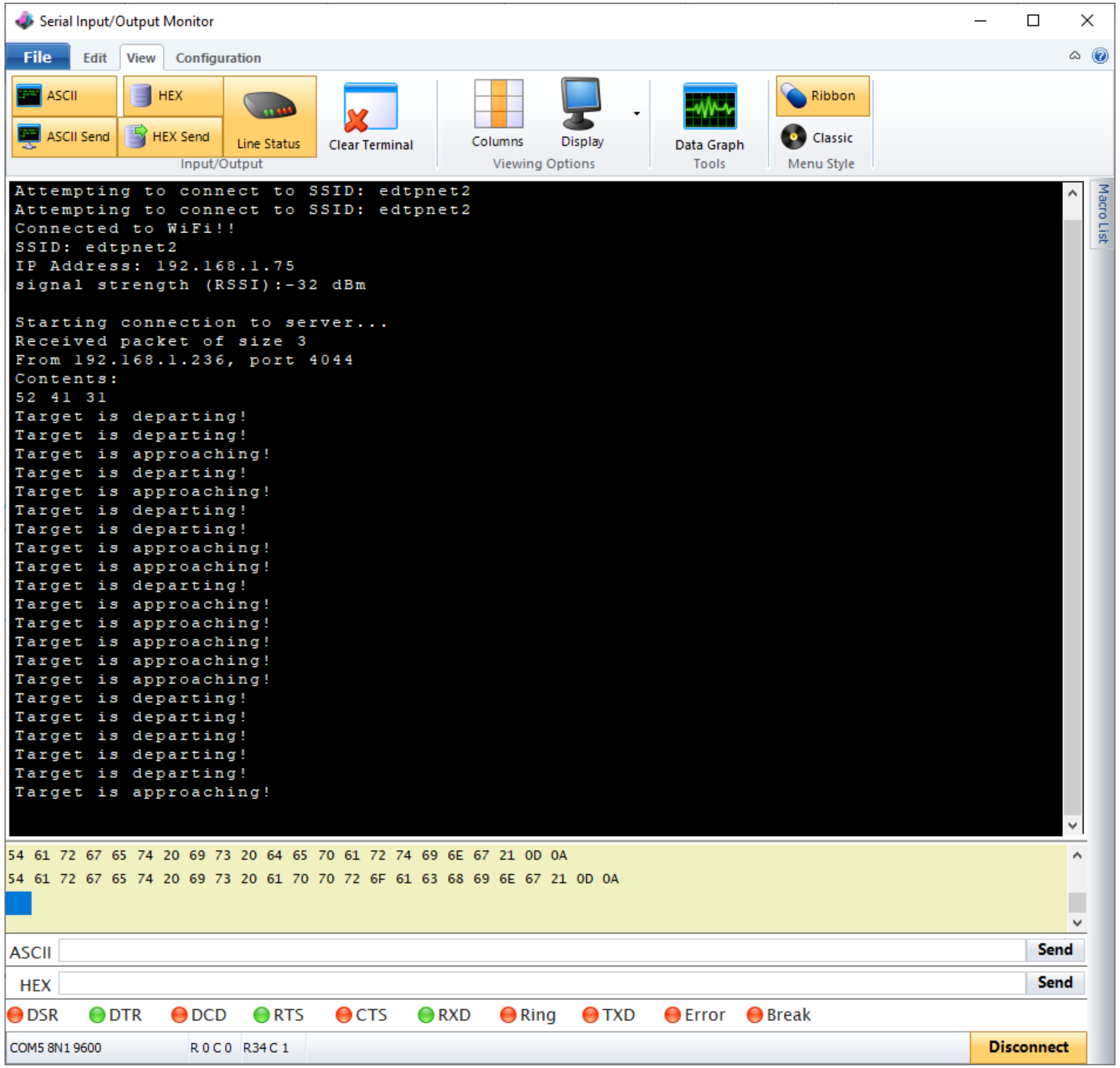
Radar Over UDP



```
void loop() {
    if (fradar == 1)
    {
        Bgt60::Direction_t direction = Bgt60::NO_DIR;
            Error_t err = radarShield.getDirection(direction);
            if (err == OK)
            {
                switch (direction)
                {
                    case Bgt60::APPROACHING:
                        sprintf(ackBuffer,"Target is approaching!\r\n");
                        Serial1.println("Target is approaching!");
                        break;
                    case Bgt60::DEPARTING:
                        sprintf(ackBuffer,"Target is departing!\r\n");
                        Serial1.println("Target is departing!");
                        break;
                    case Bgt60::NO_DIR:
                        sprintf(ackBuffer,"No Motion Detected!\r\n");
                        Serial1.println("Direction cannot be determined since no motion was detected!");
                        break;
                }
                Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
                Udp.write(ackBuffer);
                Udp.endPacket();
            }
        else{
                Serial1.println("Error occurred!");
            }
    }
```



Radar Over UDP

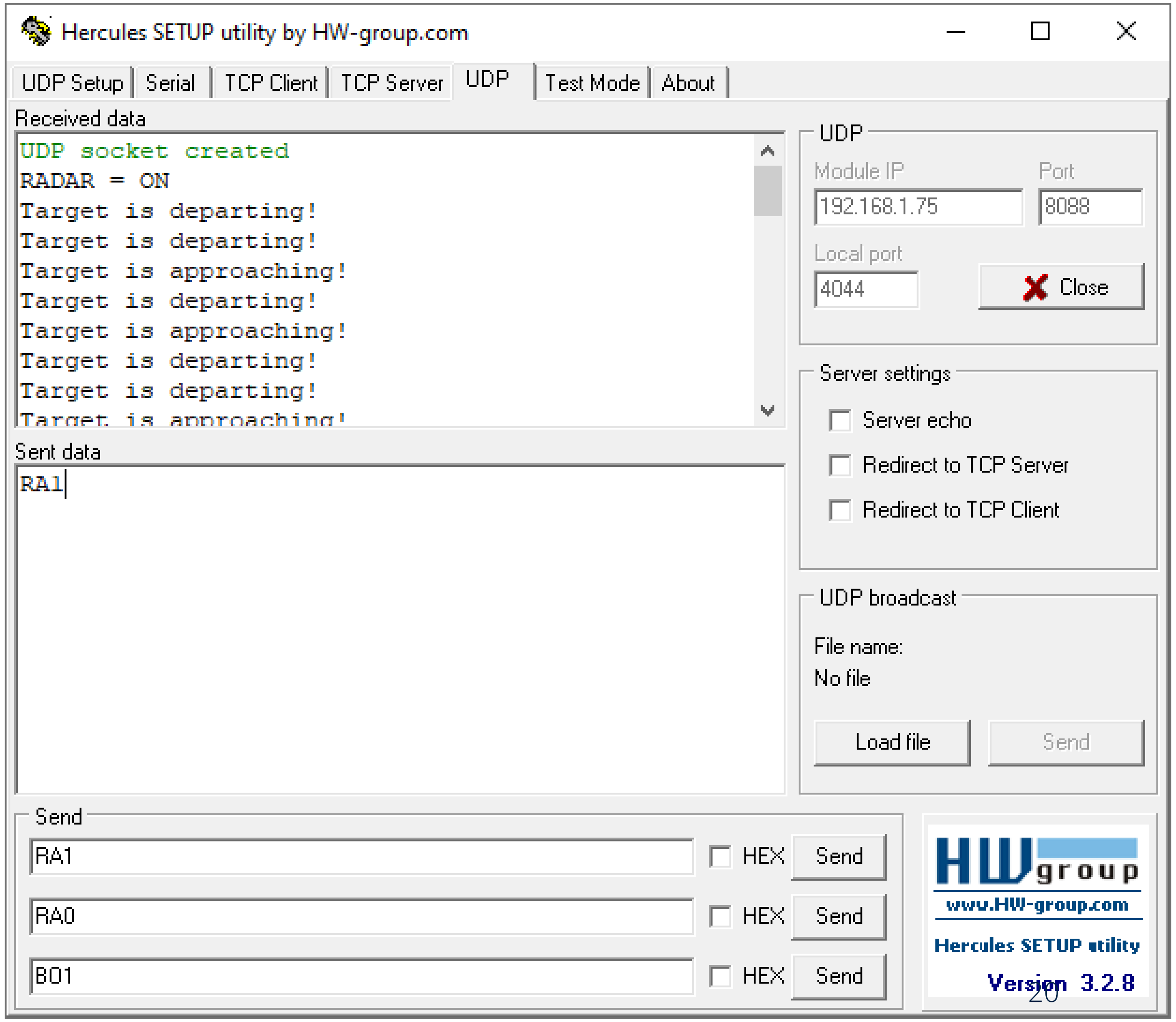


```
Attempting to connect to SSID: edtpnet2
Attempting to connect to SSID: edtpnet2
Connected to WiFi!!
SSID: edtpnet2
IP Address: 192.168.1.75
signal strength (RSSI):-32 dBm

Starting connection to server...
Received packet of size 3
From 192.168.1.236, port 4044
Contents:
52 41 31
Target is departing!
Target is departing!
Target is approaching!
Target is departing!
Target is approaching!
Target is departing!
Target is departing!
Target is approaching!
Target is approaching!
Target is departing!
Target is approaching!
Target is approaching!
Target is approaching!
Target is approaching!
Target is approaching!
Target is departing!
Target is departing!
Target is departing!
Target is departing!
Target is departing!
Target is approaching!
```

```
54 61 72 67 65 74 20 69 73 20 64 65 70 61 72 74 69 6E 67 21 0D 0A
54 61 72 67 65 74 20 69 73 20 61 70 70 72 6F 61 63 68 69 6E 67 21 0D 0A
```

ASCII Send
HEX Send

COM5 8N1 9600 R 0 C 0 R34 C 1



Received data

```
UDP socket created
RADAR = ON
Target is departing!
Target is departing!
Target is approaching!
Target is departing!
Target is approaching!
Target is departing!
Target is departing!
Target is approaching!
```

Sent data

```
RA1
```

UDP
Module IP: 192.168.1.75
Port: 8088
Local port: 4044

Server settings
- ☐ Server echo
- ☐ Redirect to TCP Server
- ☐ Redirect to TCP Client

UDP broadcast
File name:
No file

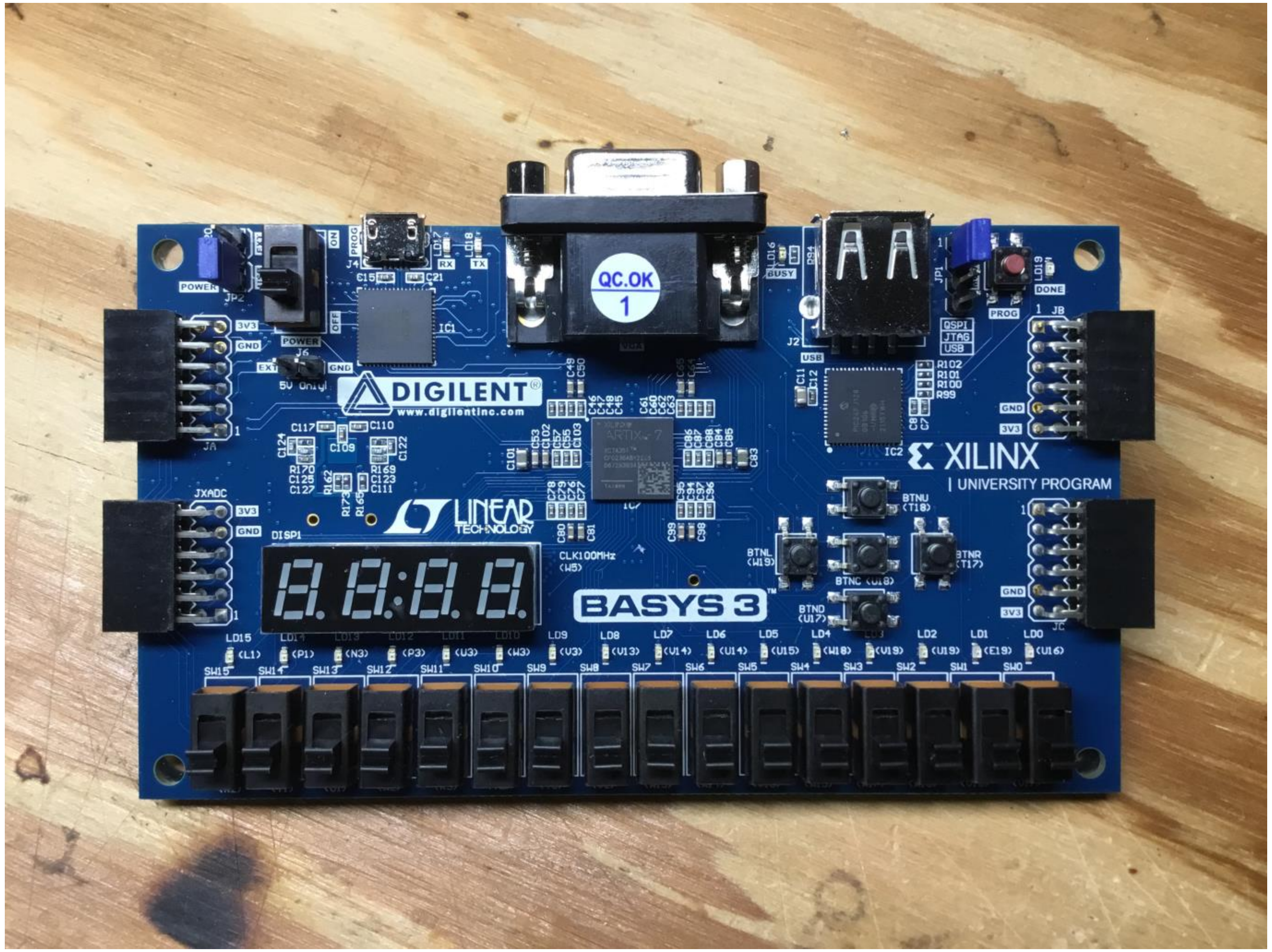
Send
- RA1 ☐ HEX Send
- RA0 ☐ HEX Send
- B01 ☐ HEX Send

Thank you for attending!!!

Please consider the resources below:

- arduino.cc
- raspberrypi.org
- infineon.com

MORE TO COME..



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

informa markets