CEC Continuing Education Center

DesignNews

Raspberry Pi 4B Application Development Using the C Programming Language

DAY 4: Coding a Raspberry Pi 4B SPI IoT Application

Sponsored by

Digi-Key ELECTRONICS

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 ‘Group Chat’ by maximizing the chat widget in your dock.
- Submit questions for the lecturer using the Q&A widget. They will follow-up after the lecture portion concludes.

Fred Eady

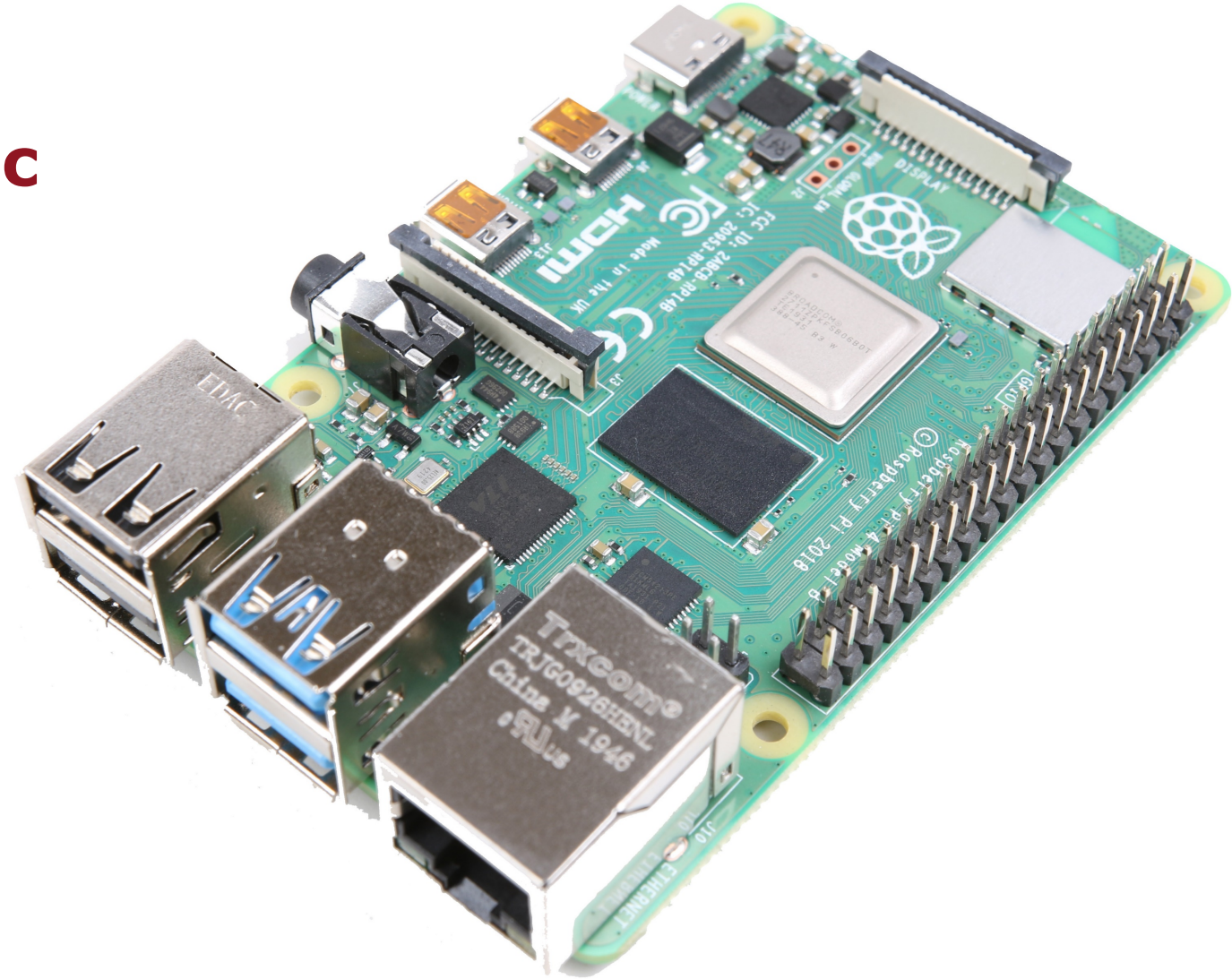
Visit ‘Lecturer Profile’ in your console for more details.

CEC Continuing Education Center

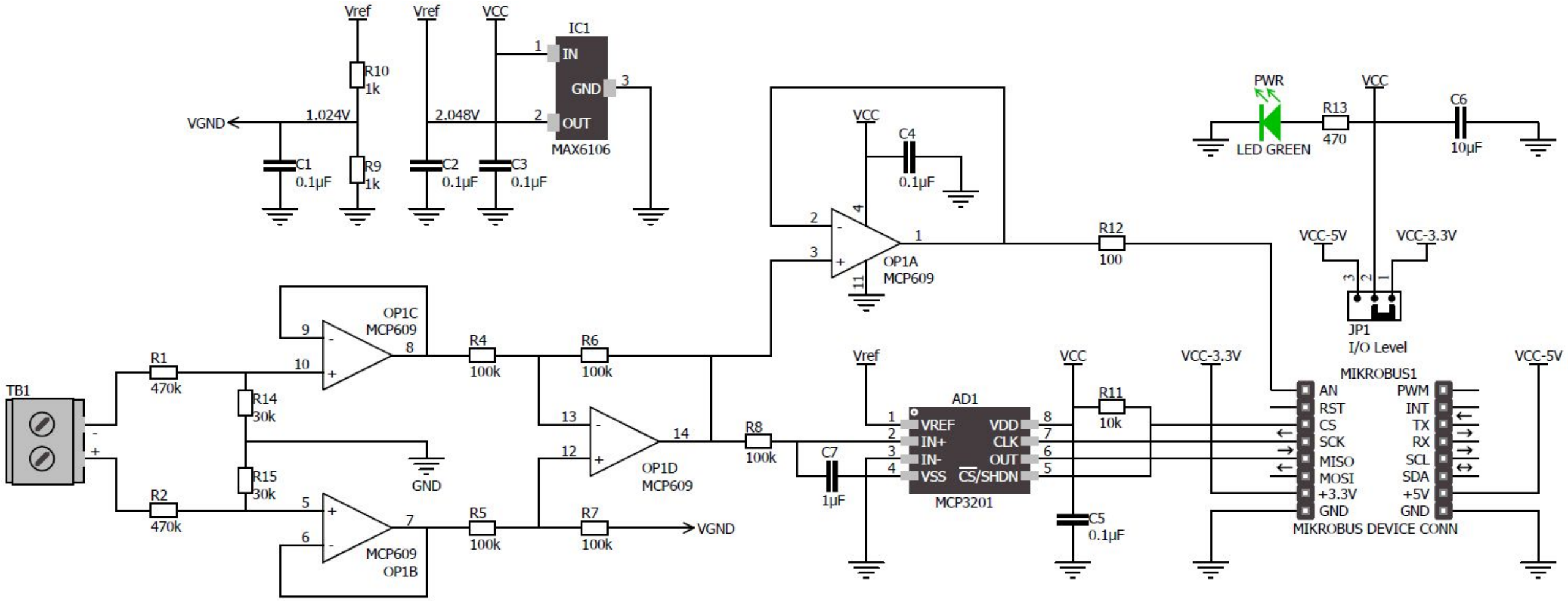
Sponsored By Digi-Key Corporation

AGENDA

- Voltmeter Circuitry
- Voltmeter Firmware Logic
- Compile and Debug
- SPI Write with pigpio

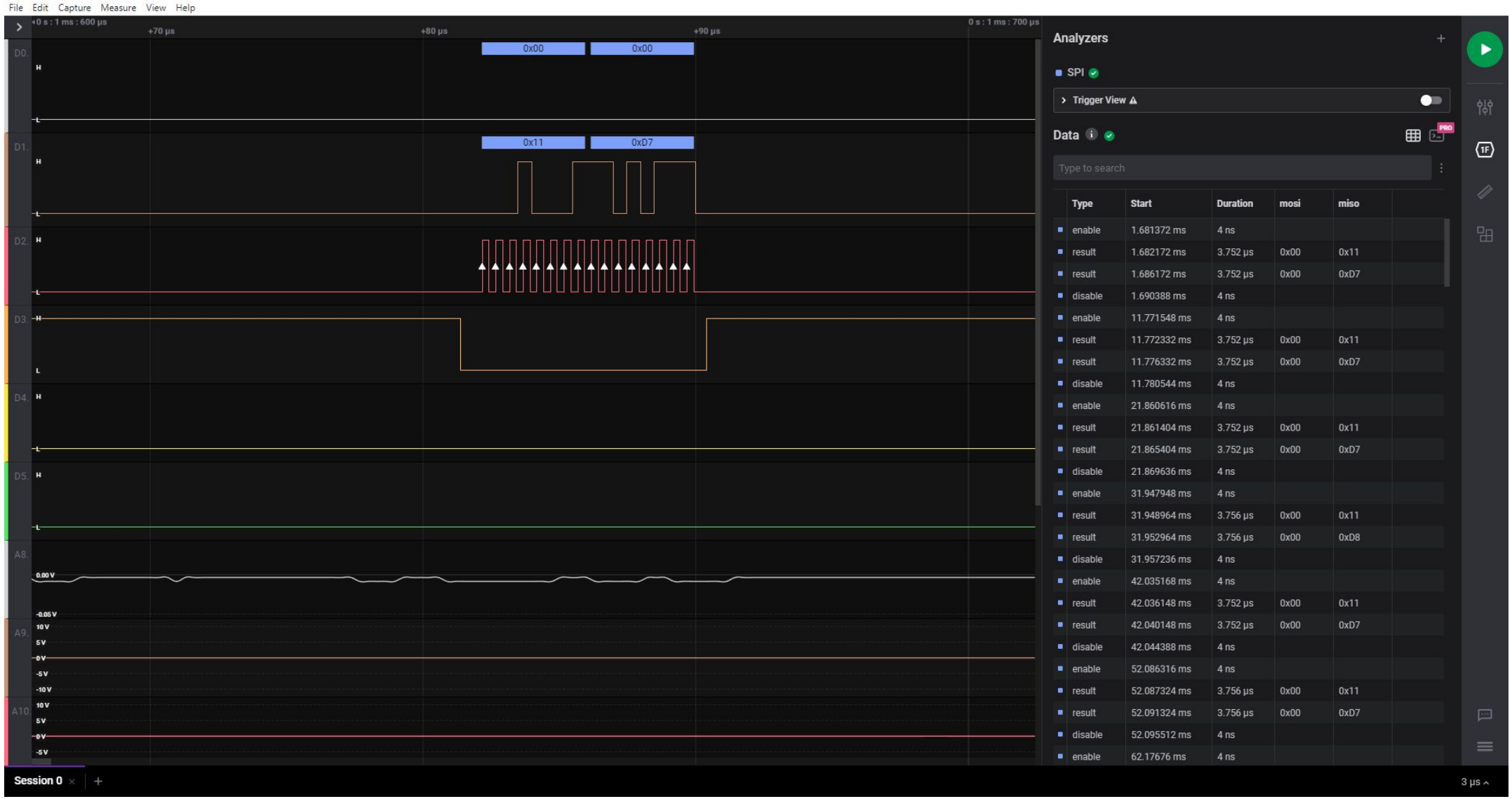


Voltmeter Circuitry



CEC Continuing Education Center

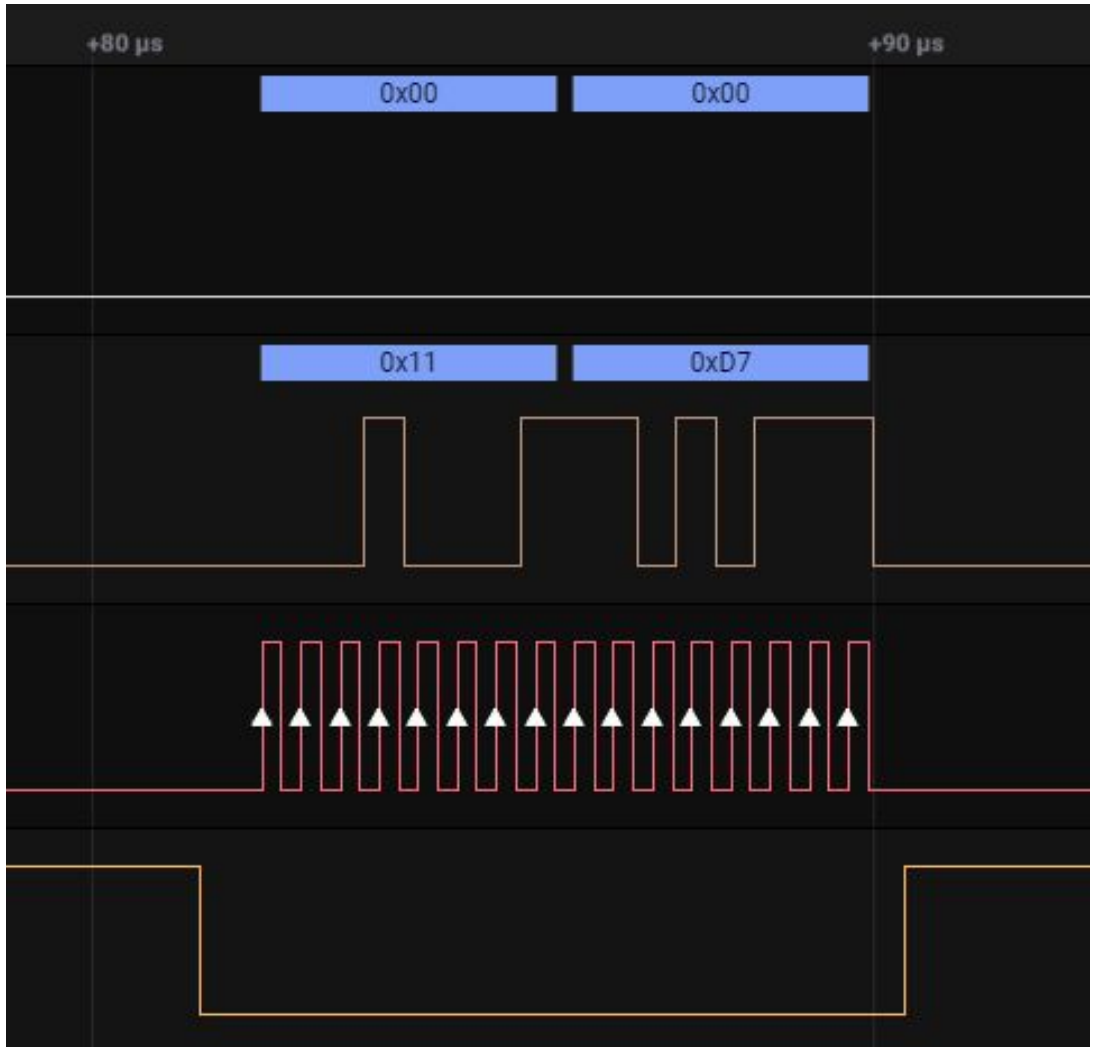
Voltmeter Firmware Logic – Extract 12 bits



CEC Continuing Education Center

Voltmeter Firmware Logic – Extract 12 bits

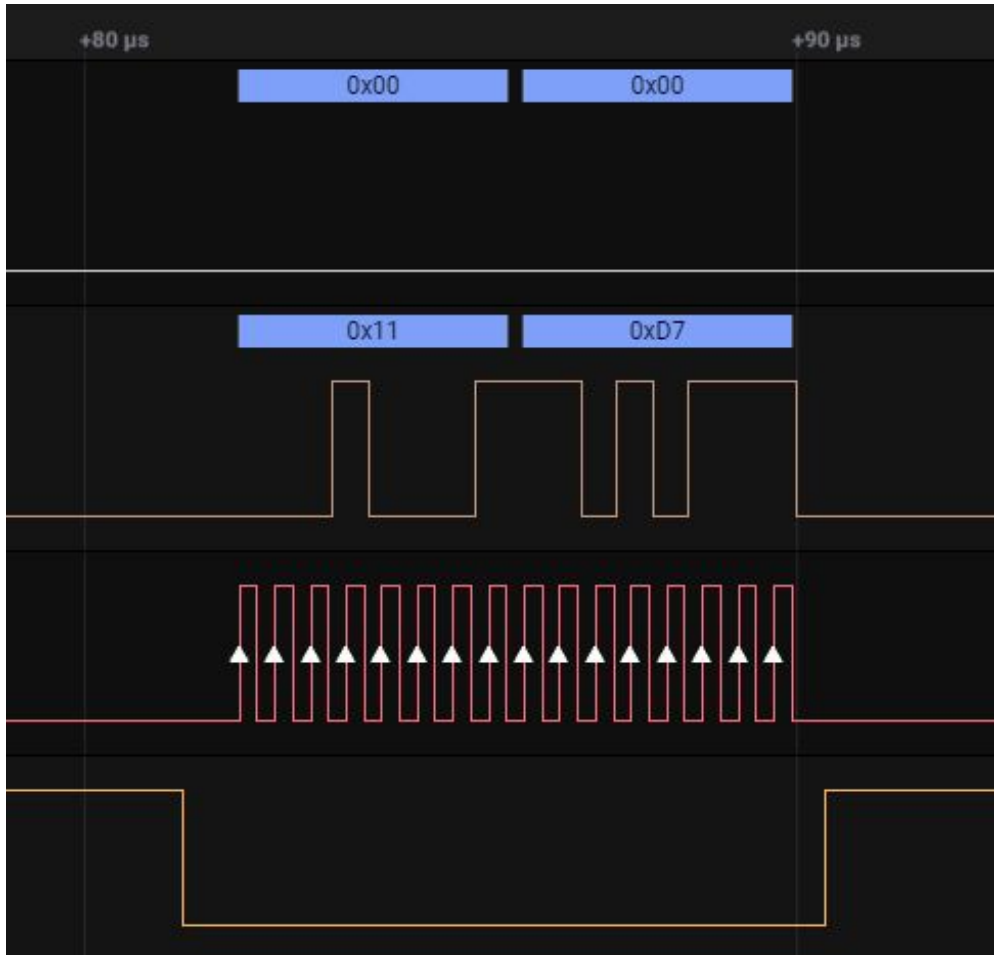
? ? 0 B11 B10 B09 B08 B07 B06 B05 B04 B03 B02 B01 B00 B00



Voltmeter Firmware Logic – Extract 12 bits

? ? 0 B11 B10 B09 B08 B07 B06 B05 B04 B03 B02 B01 B00 B00

??01 0001 1101 0111

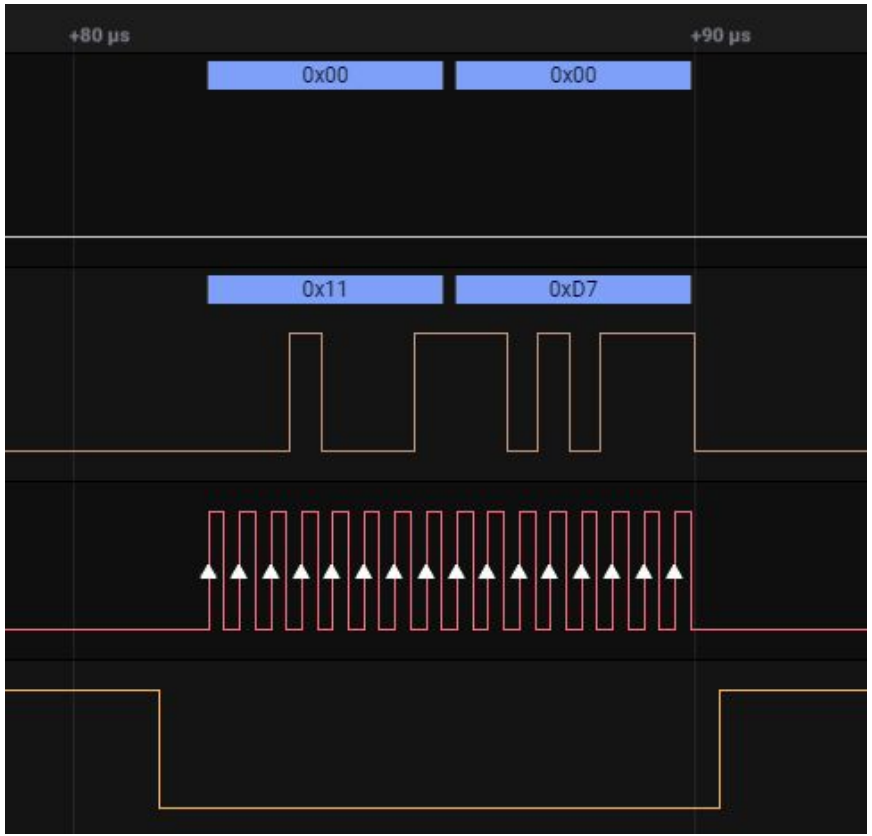


Voltmeter Firmware Logic – Extract 12 bits

? ? 0 B11 B10 B09 B08 B07 B06 B05 B04 B03 B02 B01 B00 B00

??01 0001 1101 0111

rawVoltage = make16(??010001,11010111)

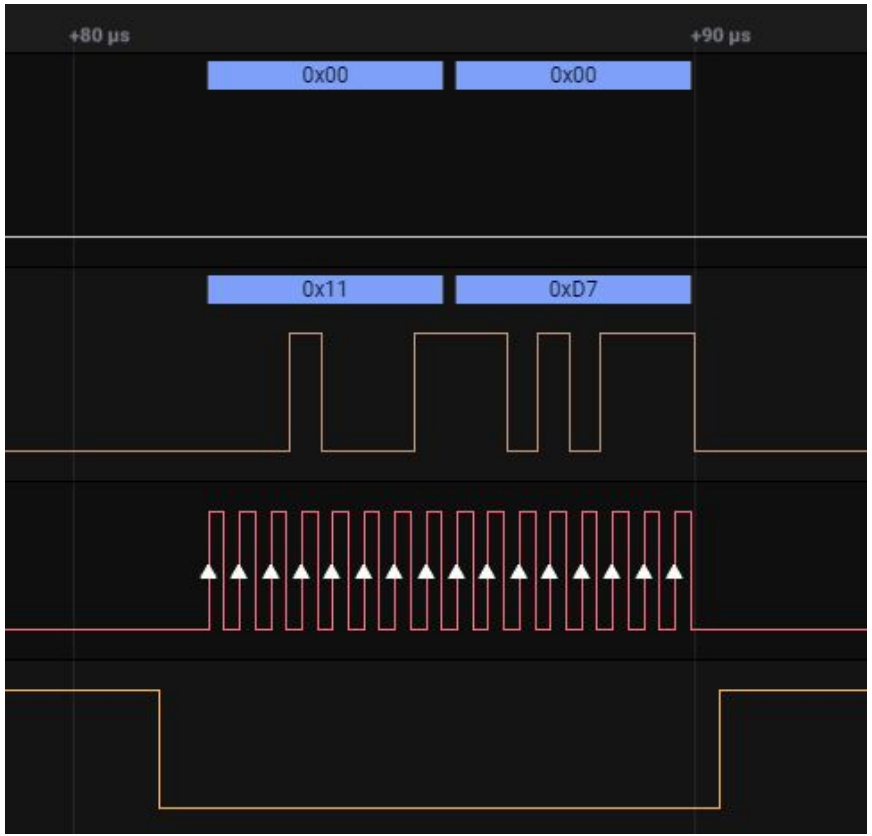


Voltmeter Firmware Logic – Extract 12 bits

? ? 0 B11 B10 B09 B08 B07 B06 B05 B04 B03 B02 B01 B00 B00

rawVoltage = 0b??0100011110101111

rawVoltage = rawVoltage >> 1

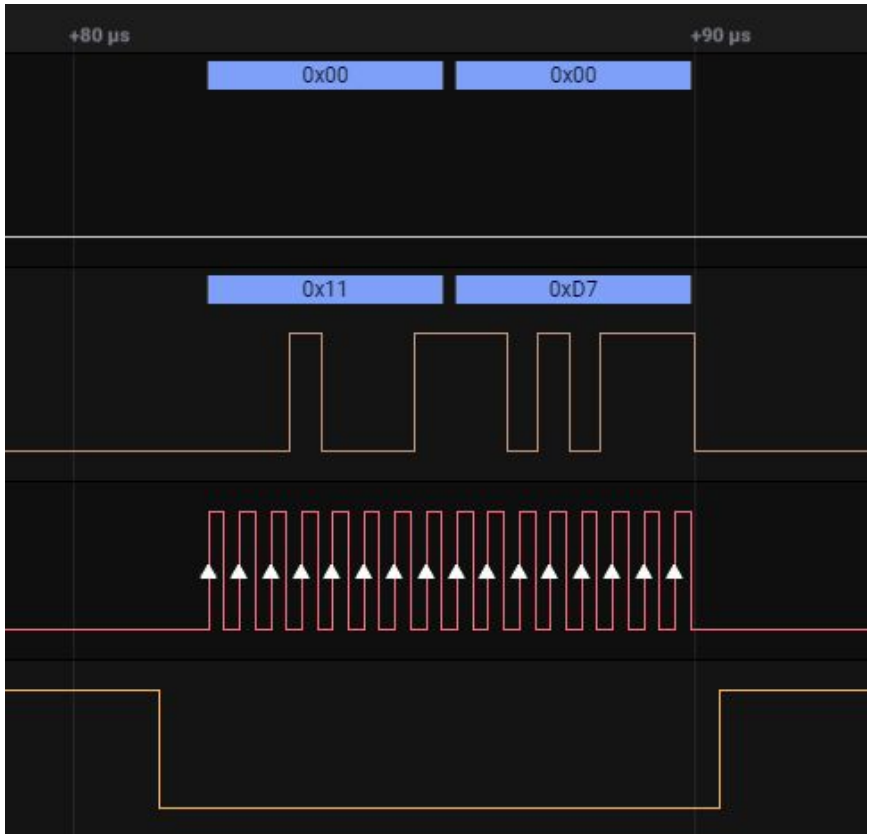


Voltmeter Firmware Logic – Extract 12 bits

? ? 0 B11 B10 B09 B08 B07 B06 B05 B04 B03 B02 B01 B00 B00

rawVoltage = 0bx??0100011101011

rawVoltage = rawVoltage & 0x0FFF

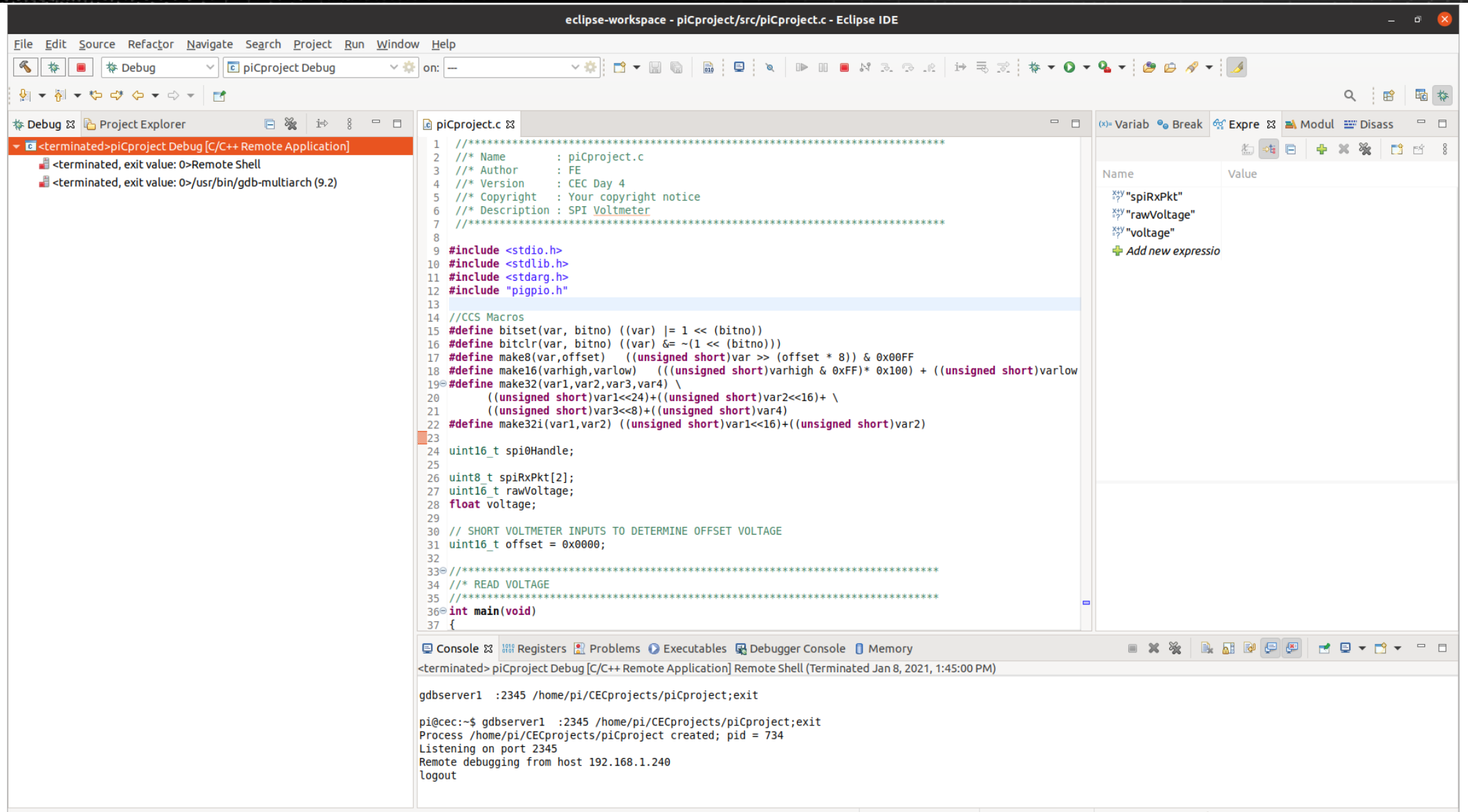


Voltmeter Firmware Logic

```
24 uint16_t spi0Handle;
25
26 uint8_t spiRxPkt[2];
27 uint16_t rawVoltage;
28 float voltage;
29
30 // SHORT VOLTMETER INPUTS TO DETERMINE OFFSET VOLTAGE
31 uint16_t offset = 0x0000;
32
33 //*****************************************************************************
34 //* READ VOLTAGE
35 //*****************************************************************************
36 int main(void)
37 {
38     if(gpioInitialise() < 0)                    //init pigpio
39     {
40         return 1;
41     }
42
43     spi0Handle = spiOpen(0,1000000,0);          //open SPI0 at 1M baud
44
45     do{
46         spiRead(spi0Handle,spiRxPkt,2);                 //read voltmeter
47         rawVoltage = make16(spiRxPkt[0],spiRxPkt[1]);   //form a 16-bit value
48         rawVoltage = rawVoltage >> 1;                   //discard extra LSB
49         rawVoltage = rawVoltage & 0x0FFF;               //eliminate 3 MSB trash bits
50         voltage = (rawVoltage - offset) * 16.6667;      //convert to millivolts
51         gpioSleep(PI_TIME_RELATIVE, 1, 0);              //delay
52     }while(1);
53 }
```

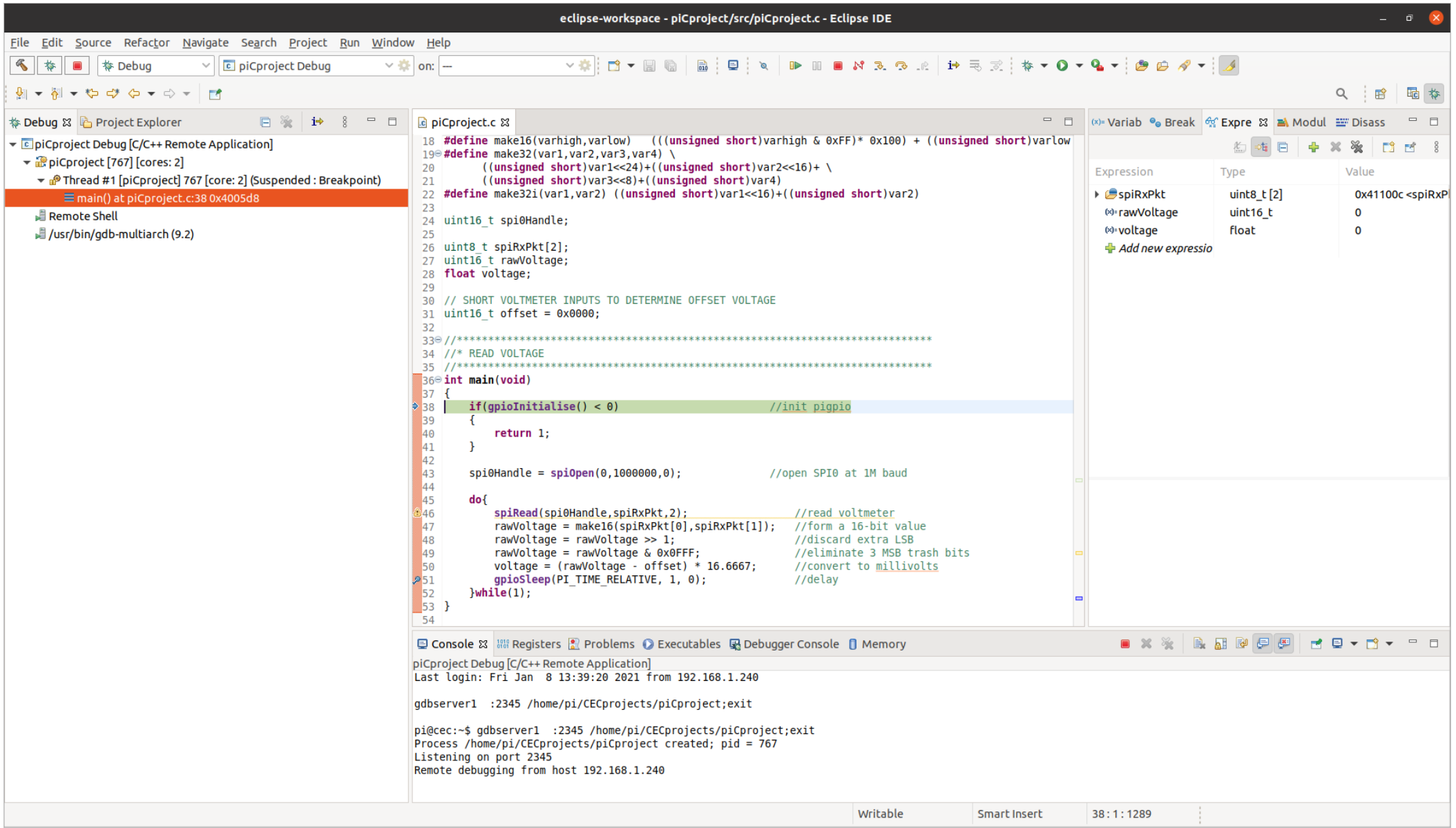
CEC Continuing Education Center

Compile and Debug - Voltmeter



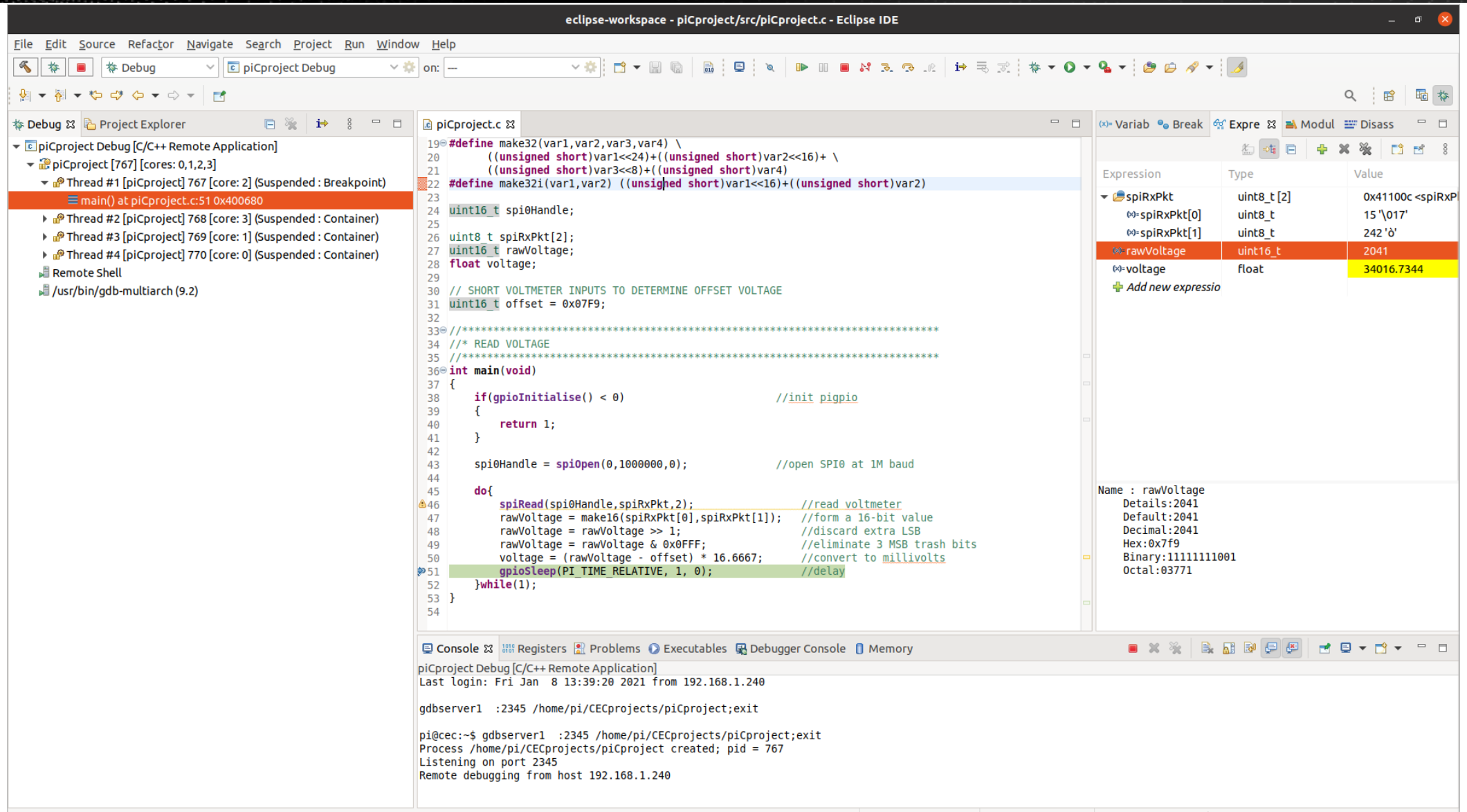
CEC Continuing Education Center

Compile and Debug - Voltmeter

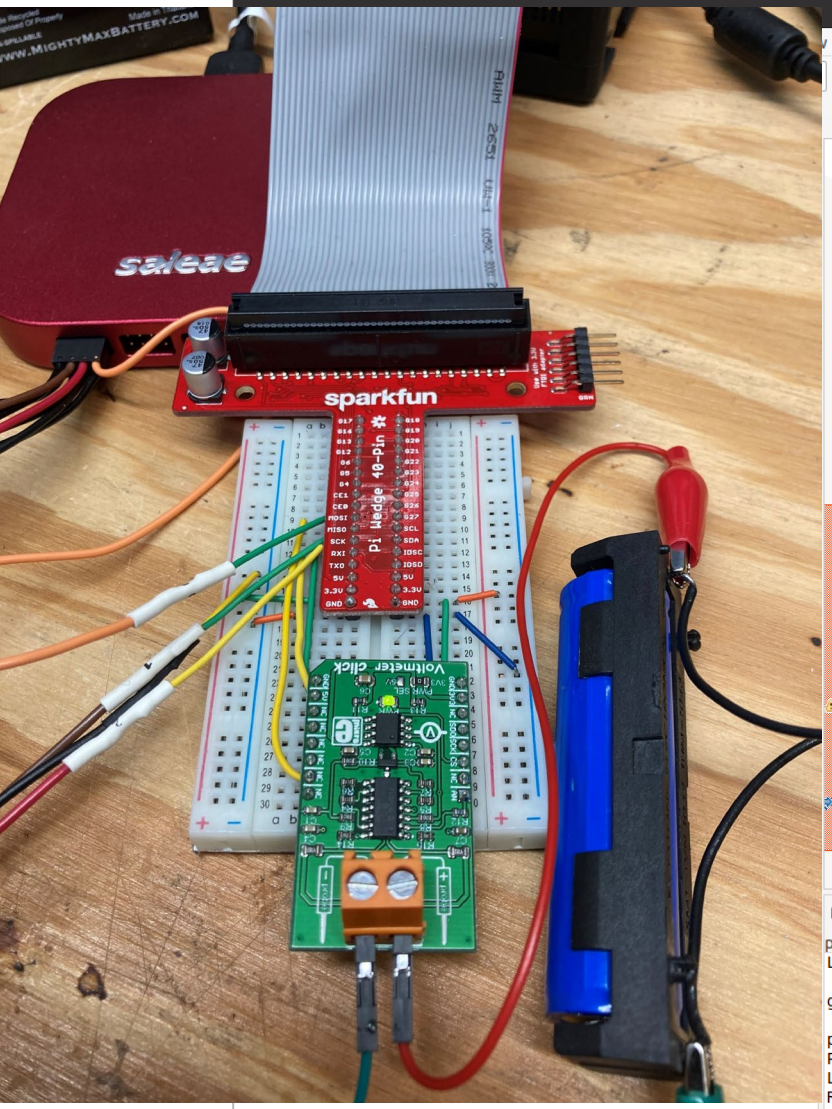


CEC Continuing Education Center

Voltmeter Firmware Logic – Offset Voltage



Compile and Debug - Voltmeter



eclipse-workspace - piCproject/src/piCproject.c - Eclipse IDE

Help

piCproject.c

```
#define make32(var1,var2,var3,var4) \
      ((unsigned short)var1<<24)+((unsigned short)var2<<16)+ \
      ((unsigned short)var3<<8)+((unsigned short)var4)
#define make32i(var1,var2) ((unsigned short)var1<<16)+((unsigned short)var2)

uint16_t spi0Handle;

uint8_t spiRxPkt[2];
uint16_t rawVoltage;
float voltage;

// SHORT VOLTMETER INPUTS TO DETERMINE OFFSET VOLTAGE
uint16_t offset = 0x07F9;

//***************************************************************************
//* READ VOLTAGE
//***************************************************************************
int main(void)
{
    if(gpioInitialise() < 0)                    //init pigpio
    {
        return 1;
    }

    spi0Handle = spiOpen(0,1000000,0);          //open SPI0 at 1M baud

    do{
        spiRead(spi0Handle,spiRxPkt,2);             //read voltmeter
        rawVoltage = make16(spiRxPkt[0],spiRxPkt[1]);   //form a 16-bit value
        rawVoltage = rawVoltage >> 1;               //discard extra LSB
        rawVoltage = rawVoltage & 0x0FFF;           //eliminate 3 MSB trash bits
        voltage = (rawVoltage - offset) * 16.6667;  //convert to millivolts
        gpioSleep(PI_TIME_RELATIVE, 1, 0);          //delay
    }while(1);
}
```

Variab | Break | Expre | Modul | Disass

Expression	Type	Value
spiRxPkt	uint8_t [2]	0x411010 <spiRxP
rawVoltage	uint16_t	2284
voltage	float	4050.00806
Add new expressio		

Console | Registers | Problems | Executables | Debugger Console | Memory

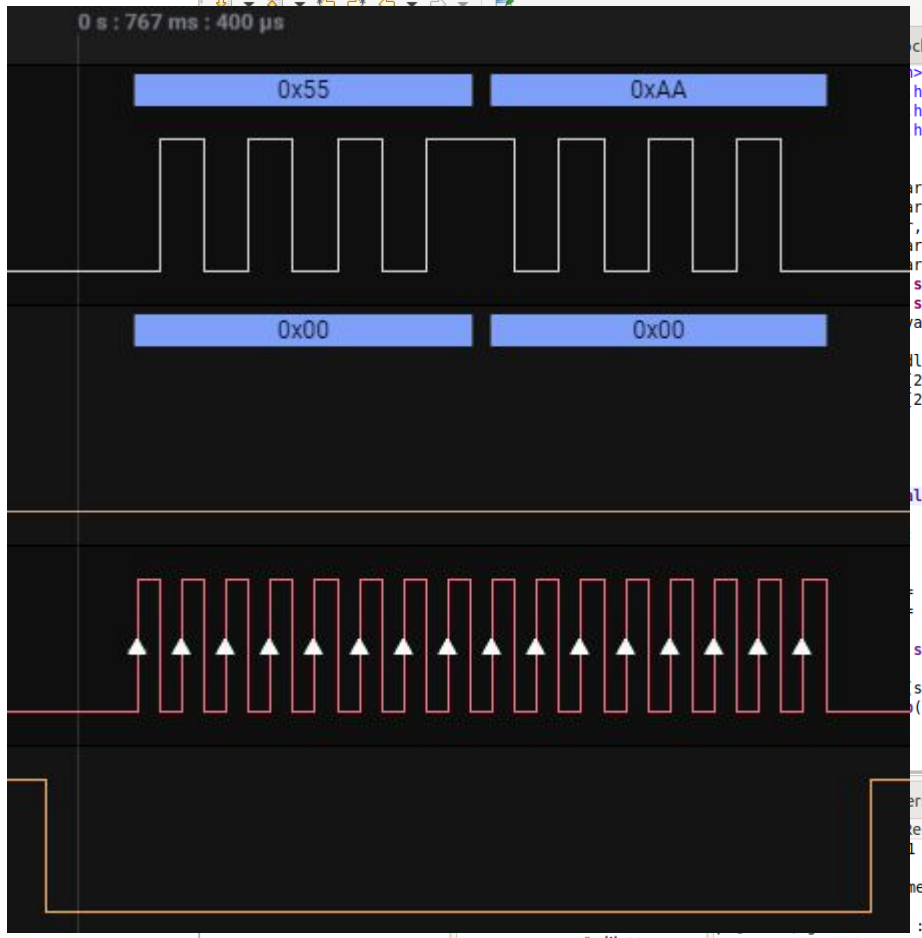
```
piCproject Debug [C/C++ Remote Application]
Last login: Fri Jan  8 13:52:36 2021 from 192.168.1.240

gdbserver1  :2345 /home/pi/CECprojects/piCproject;exit

pi@cec:~$ gdbserver1  :2345 /home/pi/CECprojects/piCproject;exit
Process /home/pi/CECprojects/piCproject created; pid = 800
Listening on port 2345
Remote debugging from host 192.168.1.240
```

Writable | Smart Insert | 51 : 1 : 1741

SPI Write – pigpio



eclipse-workspace - piCproject/src/piCproject.c - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug | piCproject Debug | on:

```
24 uint16_t spi0Handle;
25 uint8_t spiRxPkt[2];
26 uint8_t spiTxPkt[2];
27 uint8_t i;
28
29 int main(void)
30 {
31     if(gpioInitialise() < 0)                    //init pigpio
32     {
33         return 1;
34     }
35
36     spiTxPkt[0] = 0x55;
37     spiTxPkt[1] = 0xAA;
38
39     spi0Handle = spiOpen(0,1000000,0);          //open SPI0 at 1M baud
40     do{
41         spiWrite(spi0Handle,spiTxPkt,2);
42         gpioSleep(PI_TIME_RELATIVE, 1, 0);      //delay
43     }while(1);
44 }
```

Expression	Type	Value
spiTxPkt	uint8_t [2]	0x411010 <spiTx
spiTxPkt[0]	uint8_t	85 'U'
spiTxPkt[1]	uint8_t	170 'ª'
Add new expressio		

```
Process /home/pi/CECprojects/piCproject created; pid = 683
Listening on port 2345
Remote debugging from host 192.168.1.240
```

lib64 libexec libx32 local

Writable | Smart Insert | 31 : 1 : 1028

Thank you for attending

Please consider the resources below:

- https://www.raspberrypi.org
- https://ubuntu.com
- https://www.eclipse.org
- https://mikroe.com

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets