

This is the TDC1000-C2000EVM board that I am working with. The J6 connector can only hook up to the 8 GPIOs that I mentioned via U7. If I wanted to connect to other GPIOs I would have to directly solder to TMS320F28035 microcontroller correct? If so, that sounds near impossible with the tools I have.

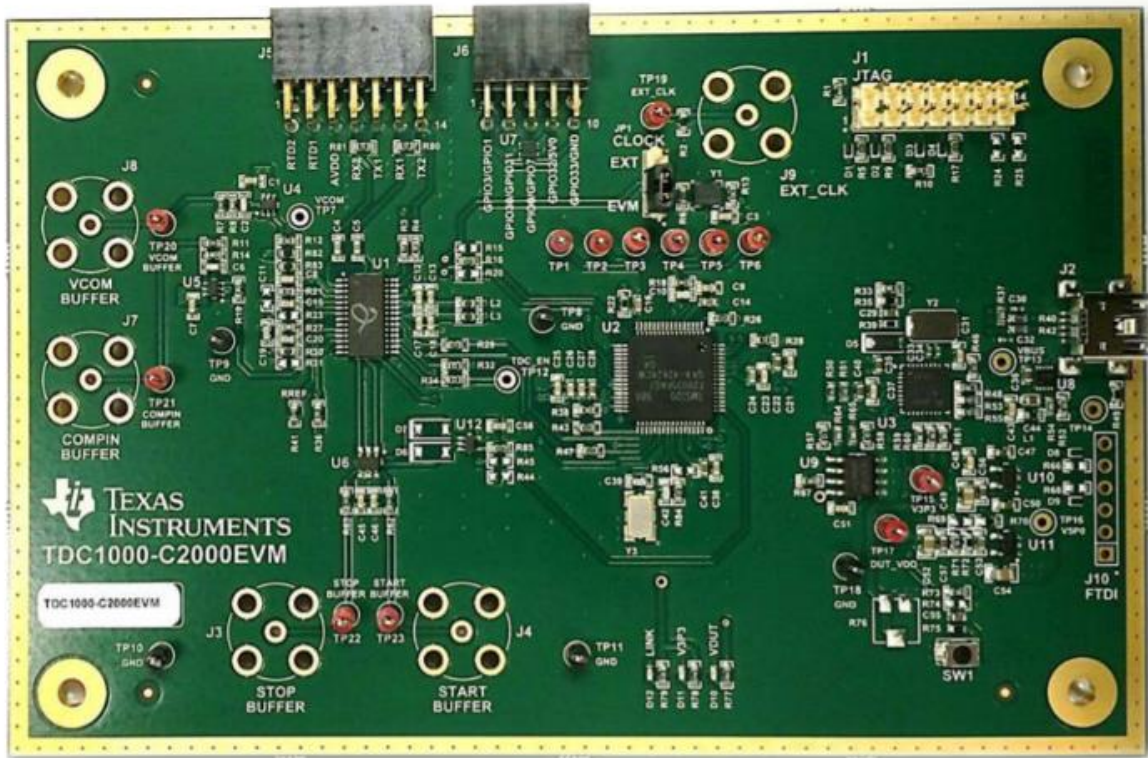
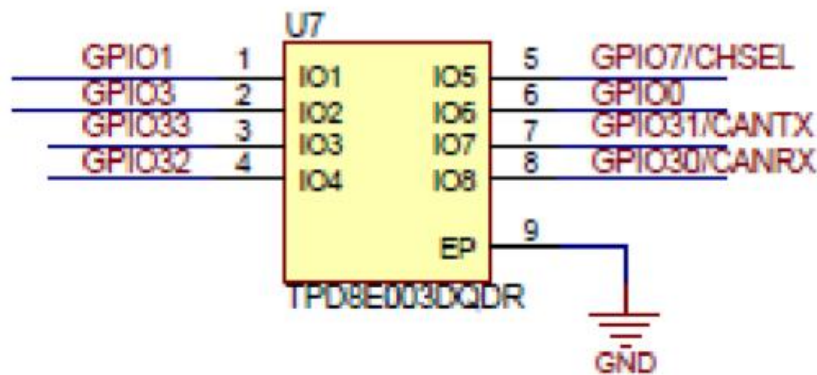


Figure 1. TDC1000-C2000EVM Evaluation Board



So I then went to the TMS320F2803x piccolo reference manual:

<https://www.ti.com/lit/ug/sprui10/sprui10.pdf>

From the manual I looked up the available GPIOs to see what forms of communication are available. Depending on what value is written to the bit, it will change what the GPIO can do. Pretty self-explanatory.

7-6	GPIO3		Configure the GPIO3 pin as: 00 GPIO3 - General purpose I/O 3 (default) (I/O) 01 EPWM2B - ePWM2 output B (O) 10 SPISOMIA (I/O) - SPI-A Slave output/Master input 11 COMP2OUT (O) - Comparator 2 output
5-4	GPIO2		Configure the GPIO2 pin as: 00 GPIO2 (I/O) General purpose I/O 2 (default) (I/O) 01 EPWM2A - ePWM2 output A (O) 10 Reserved. 11 Reserved.
3-2	GPIO1		Configure the GPIO1 pin as: 00 GPIO1 - General purpose I/O 1 (default) (I/O) 01 EPWM1B - ePWM1 output B (O) 10 Reserved 11 COMP1OUT (O) - Comparator 1 output
1-0	GPIO0		Configure the GPIO0 pin as: 00 GPIO0 - General purpose I/O 0 (default) (I/O) 01 EPWM1A - ePWM1 output A (O) 10 Reserved. 11 Reserved.

So then I found the example code of the GPIOs and modified it to fit my project. Below is a snippet of my code:

```

92 void
93 Gpio_setup(void)
94 {
95     //
96     // Enable a GPIO output on GPIO0
97     //
98     GpioCtrlRegs.GPAPUD.bit.GPIO0 = 0; // Enable pullup on GPIO0
99     GpioDataRegs.GPASET.bit.GPIO0 = 1; // Load output latch
100    GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0; // GPIO0 = GPIO0
101    GpioCtrlRegs.GPADIR.bit.GPIO0 = 1; // GPIO0 = ePWM1 output A output
102
103    //
104    // Enable a GPIO output on GPIO1
105    //
106    GpioCtrlRegs.GPAPUD.bit.GPIO1 = 0; // Enable pullup on GPIO1
107    GpioDataRegs.GPASET.bit.GPIO1 = 1; // Load output latch
108    GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 0; // GPIO1 = GPIO1
109    GpioCtrlRegs.GPADIR.bit.GPIO1 = 1; // GPIO1 = ePWM1 output B output
110
111    //
112    // Enable a GPIO output on GPIO3
113    //
114    GpioCtrlRegs.GPAPUD.bit.GPIO3 = 0; // Enable pullup on GPIO3
115    GpioDataRegs.GPASET.bit.GPIO3 = 1; // Load output latch
116    GpioCtrlRegs.GPAMUX1.bit.GPIO3 = 0; // GPIO3 = GPIO3
117    GpioCtrlRegs.GPADIR.bit.GPIO3 = 1; // GPIO1 = ePWM2 output B output
118
119    //
120    // Enable a GPIO output on GPIO7
121    //
122    GpioCtrlRegs.GPAPUD.bit.GPIO7 = 0; // Enable pullup on GPIO7
123    GpioDataRegs.GPASET.bit.GPIO7 = 1; // Load output latch

```

Problems 0 items

Description	Resource	Path	Location	Type

In the code, I am flipping the GPIOs to outputs and then making them the specified outputs that they have. For GPIO0, I set it to 1 for the EPWM1A – ePWM1 output a (O) setting.

Also, in your previous response I got the terminology wrong and said read outputs. I meant that I want to read the values of the outputs that they are writing. So yes I am trying to write the outputs. Once I get these values though I don't know where to write them to convert the value so that's why I said I would like to send the values to an Arduino to see these values and convert them to their original value of uS.

Essentially my plan was to write the outputs from the TDC board to an Arduino. Read the values on the Arduino and then create an if statement for when the sensors read a certain value. Once that if statement were complete it would then send a digital output to a PLC to shut a valve. Do you think this is the best course of action to take or are there better ways to go about this?

On another note, I am using the XDS110 Debug probe to connect to the board via J1 and debug it with the code above.

<https://www.ti.com/tool/TMDSEMU110-U>

I was told this is what to use but just wanted to make sure that this is the correct way to go about this.

