Due on 10/13, submit a single file as doc, docx, or pdf in Canvas

1. (20%)

Define a union data struct WORD\_T for a uint16\_t integer so that a value can be assigned to a WORD\_T integer in three ways:

(1) To assign the value to each bit of the integer,

(2) To assign the value to each byte of the integer,

(3) To assign the value to the integer directly.

a) Show the code of defining the union data struct WORD\_T.

b) Show the code to assign 17 to the WORD\_T variable foo.

b.1) Assign 17 to the bits of foo.

b.2) Assign 17 to the bytes of foo.

b.3) Assign 17 to the integer of foo.

Hint: You may refer to the source file ${xc16}/support/PIC24E/h/p24EP512GU810.h. ${xc16} is where you installed the xc16 compiler.

=== code ===

#typedef union \_WORD\_T {

|  |  |
| --- | --- |
| struct { |  |
|  | unsigned B0: 1; |
|  | unsigned B1: 1; |
|  | unsigned B2: 1; |
|  | unsigned B3: 1; |
|  | unsigned B4: 1; |
|  | unsigned B5: 1; |
|  | unsigned B6: 1; |
|  | unsigned B7: 1; |
|  | unsigned B8: 1; |
|  | unsigned B9: 1; |
|  | unsigned B10: 1; |
|  | unsigned B11: 1; |
|  | unsigned B12: 1; |
|  | unsigned B13: 1; |
|  | unsigned B14: 1; |
|  | unsigned B15: 1; |
|  | }; |
|  | struct { |
|  | uint8\_t LB; |
|  | uint8\_t HB; |
|  | }; |
|  | uint16\_t val; |

} WORD\_T;

WORD\_T foo;

b.1) foo.bit\_handler11:1

foo.bit\_handler11:0

foo.bit\_handler11:0

foo.bit\_handler11:0

foo.bit\_handler11:1

b2) int num\_17 = 0x11;

Int size\_t;

for(size\_t; i< 2;i++)

foo.value\_handler[byte(num\_17)][i];

b3) foo.int\_handler = 17

=== end of code ===

2. (10%)

Read the following code.

(a) Add one line of C code that defines a macro, so that "main()" will call "foowithdebug()".

#define DEBUG

(b) Read gcc command options about macro at https://gcc.gnu.org/onlinedocs/gcc/Preprocessor-Options.html

Show the screenshot where you add a proper gcc macro option in MPLab X, so that "main()" will call "foowithdebug()".

=== code ===

#define DEBUG

int foowithdebug() { printf("foo with debug\n");}

int foo() { printf("called foo"); }

int main() {

#ifdef DEBUG

foowithdebug();

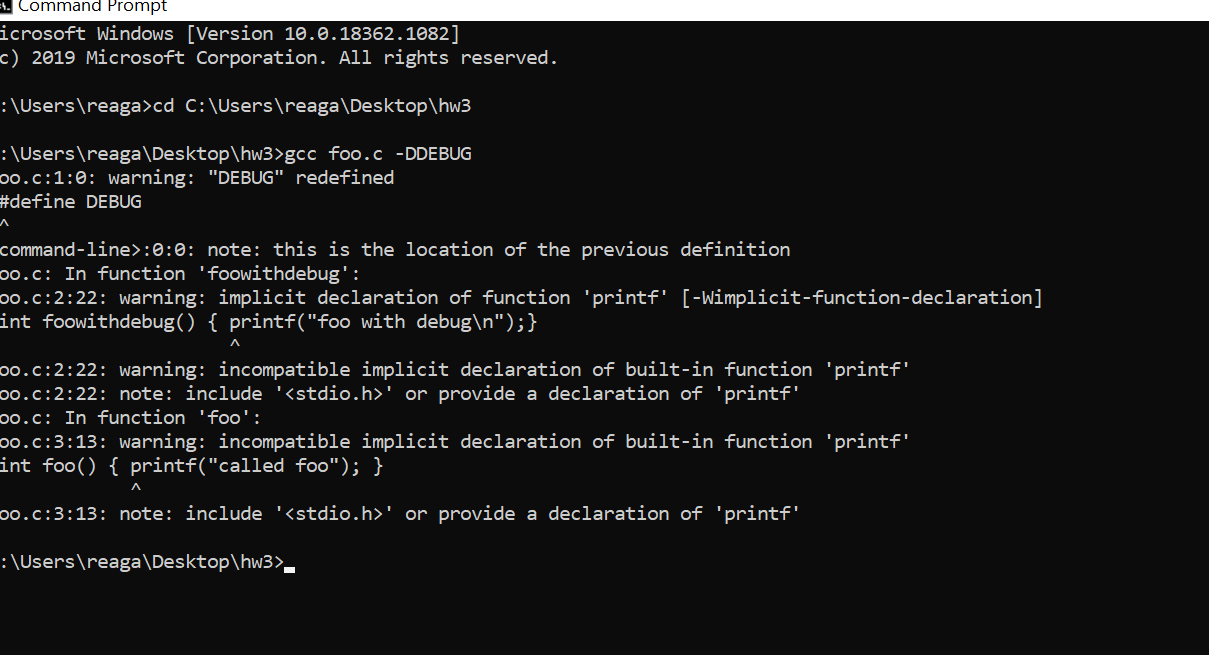
#else

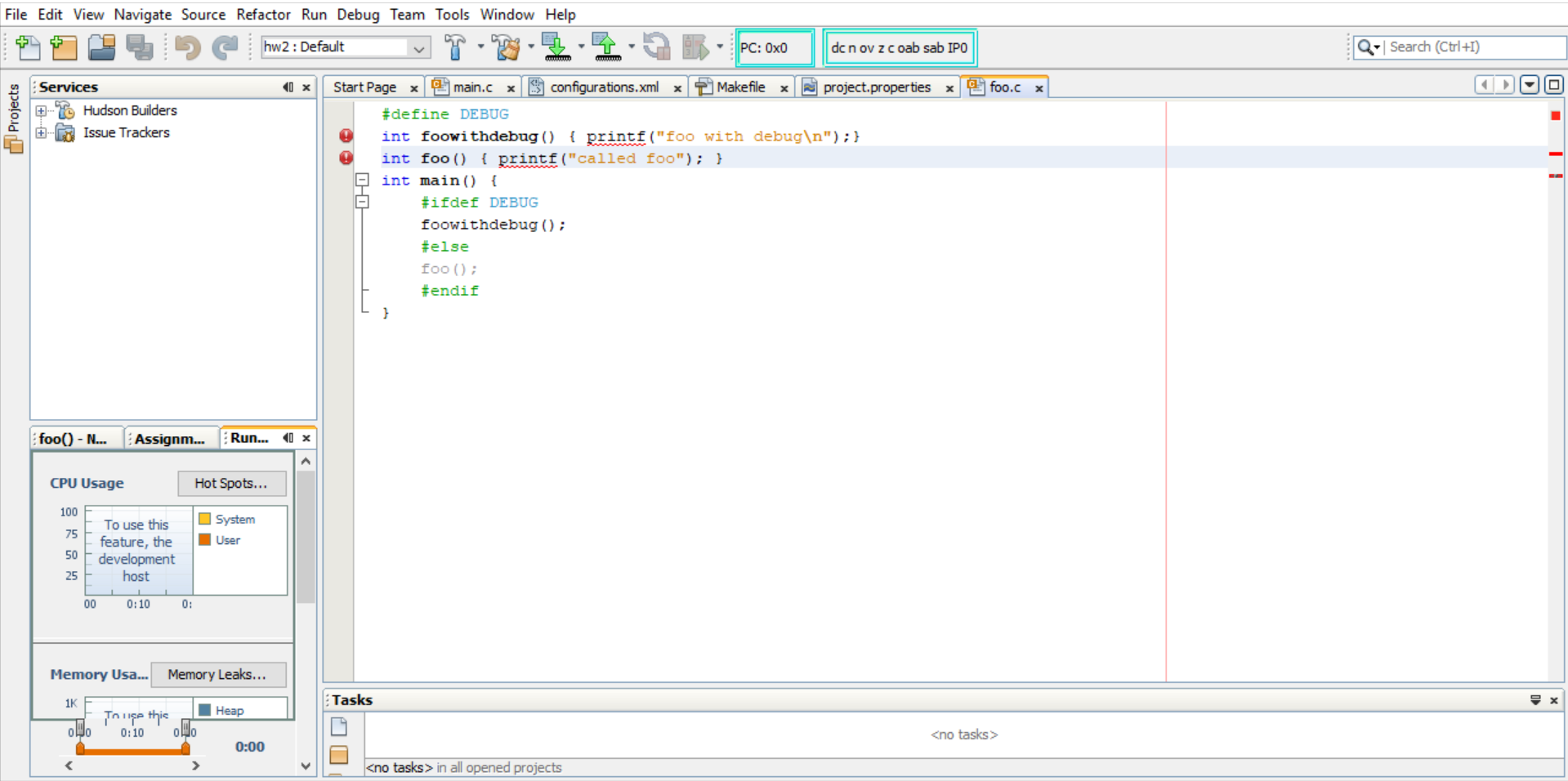
foo();

#endif

}

=== end of code ===





3. (15%)

Briefly explain what are the functions declared in each of the header files: stdlib.h, string.h, math.h, ctype.h? (Do NOT list and explain each function.)

The stdlib.h header defines four variable types,5 macros,and 28 function libraries that are used for performing general functions.

The string.h header defines one variable type, one macro, and 22 library functions that are used for manipulating arrays of characters.

The math.h header defines 21 library functions and 1 macro. All the functions available in this library take double as an argument and return double as the result.

The ctype.h header defines several functions that are useful for testing and mapping characters. All the functions accept int as a parameter whose value must be EOF or represented as an unsigned char. All the functions return non-zero if the argument c satisfies the condition, and zero if not.

4. (15%)

Read the following code.

a) Draw a memory layout to show the relations of a, b and c.

char a[3][10] = {"abcdefg", "1234567", "!@#$%^&"};

a will be base address 100

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | a  (100) | b  (101) | c  (102) | d  (103) | e  (104) | f  (105) | g  (106) | (107) | (108) | (109) |
| 1 | 1  (110) | 2  (111) | 3  (112) | 4  (113) | 5  (114) | 6  (115) | 7  (116) | (117) | (118) | (119) |
| 2 | !  (120) | @  (121) | #  (122) | $  (123) | %  (124) | ^  (125) | &  (126) | (127) | (128) | (129) |

b) What are the value in x and y?

x has the value d

y has the value 4

=== code ===

char a[3][10] = {"abcdefg", "1234567", "!@#$%^&"};

char\* b[3];

char\*\* c;

b[0] = &a[0][0];

b[1] = &a[1][0];

b[2] = &a[2][0];

c = b;

char x = b[0][3];

char y = b[0][13];

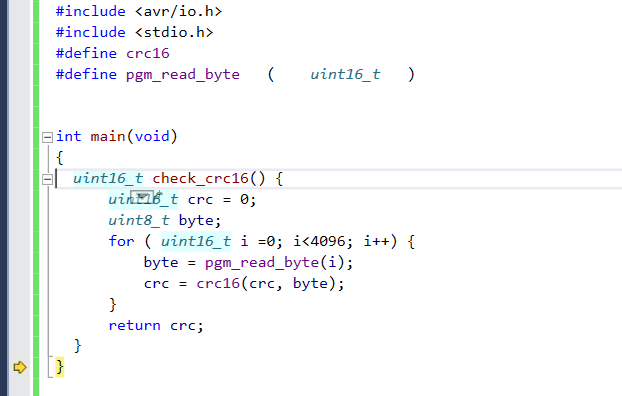
=== end of code ===

5. (20%)

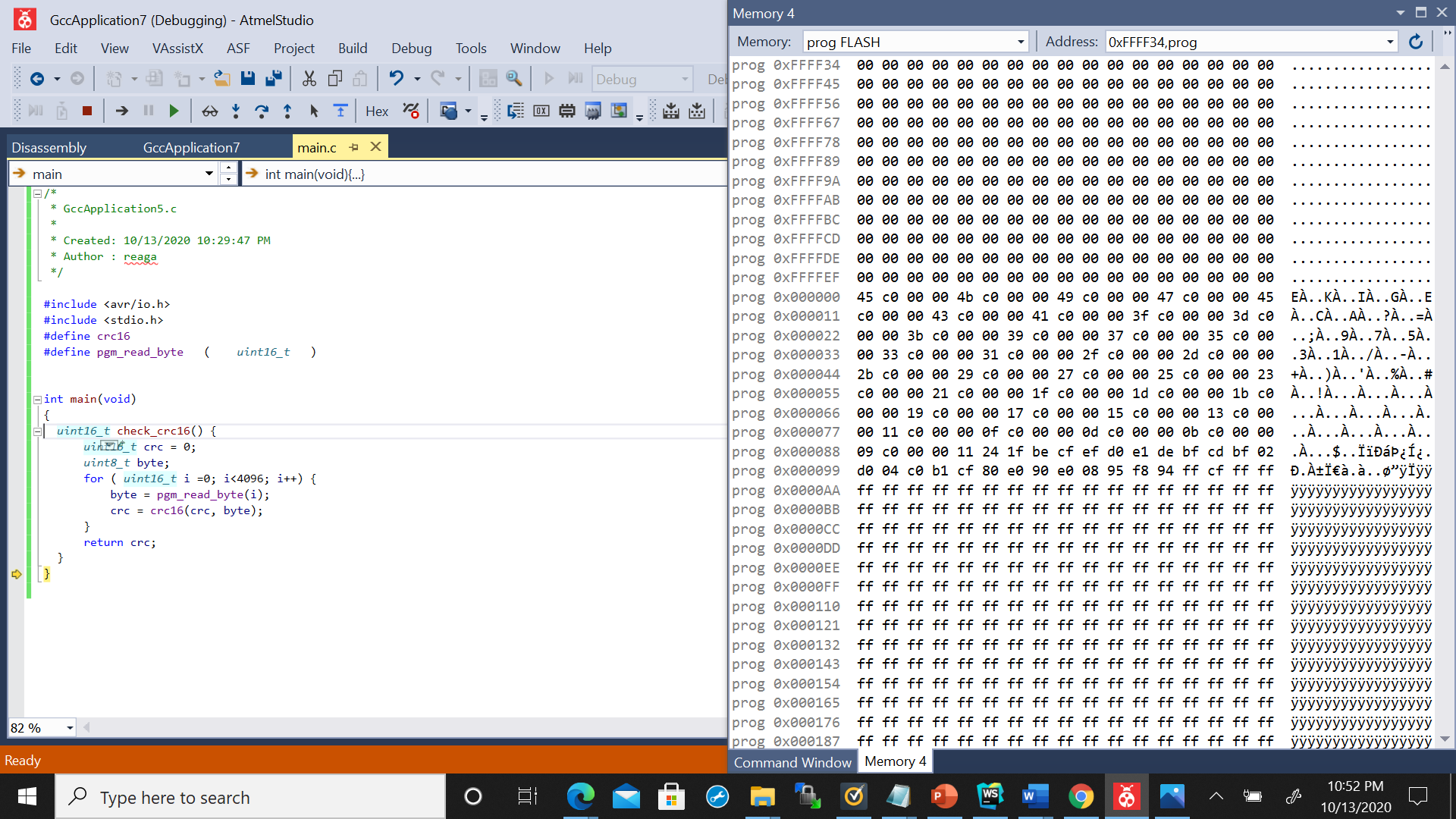
Read the reference at http://www.nongnu.org/avr-libc/user-manual/pgmspace.html and http://www.nongnu.org/avr-libc/user-manual/group\_\_util\_\_crc.html

Make a program to calculate CRC-16 of the code in program memory. Assume the code size is always 4KBytes.

a) Copy & paset your code in report.



b) Run your program in AVR Studio and show the CRC value of your program in a screen shot of AVR Studio.



6. (20%)

Read the schematics and datasheets of Atmega128 and Pic24e chips. The two schematics (hw3.sch.atmega.pdf and hw3.sch.pic.pdf) are included, and the datasheets are in canvas/course/pages/resources.

Find three types of pins: (i) reset, (ii) clock and (iii) the IO pins in ports A.

a) Document the three types of pins of Atmega128 in a table. The example table "hw3.pins.atmega.docx" is included.

|  |  |  |  |
| --- | --- | --- | --- |
| **Atmega128** | | | |
| **Pin#** | **Description** | | **Label** |
| **In Datasheet** | **In Schematic** |
| Reset | | | |
| 20 | ~RESET | RST | RSTN |
| Clock | | | |
| 24 | XTAL1 | XTAL1 | NC |
| 23 | XTAL2 | XTAL2 | NC |
|  |  |  |  |
|  |  |  |  |
| IO Pins | | | |
| 51 | PA0 (AD0) | PA0/AD0 | LED3 |
| 50 | PA1 (AD1) | PA1/AD1 | LED2 |
| 49 | PA2 (AD2) | PA2/AD2 | LED1 |
| 48 | PA3 (AD3) | PA3/AD3 | FLASH\_CS |
| 47 | PA4 (AD4) | PA4/AD4 | SERIAL\_ID |
| 46 | PA5 (AD5) | PA5/AD5 | BAT\_ON |
| 45 | PA6 (AD6) | PA6/AD6 | CHP\_OUT |
| 44 | PA7 (AD7) | PA7/AD7 | THERM\_PWR |
| 10 | PB0 (SS) | PB0/SS | NC |
| 11 | PB1 (SCK) | PB1/SCK | SPI\_SCK |
| 12 | PB2 (MOSI) | PB2/MOSI | SPI\_MOSI |
| 13 | PB3 (MISO) | PB3/MISO | SPI\_MISO |
| 14 | PB4 (OC0) | PB4/OC0 | PWM0 |
| 15 | PB5 (OC1A) | PB5/OC1A | PWM1A |
| 16 | PB6 (OC1B) | PB6/OC1B | PWM1B |
| 17 | PB7 (OC1C) | PB7/OC1C | SPI\_SCK |

**Document the IO pins in ports A.**

**If an IO pin is not connected, label the pin as NC.**

b) Document the three types of pins of Pic24e in a table. The example table "hw3.pins.pic.docx"is included.

|  |  |  |  |
| --- | --- | --- | --- |
| **Pic24ep512gu810** | | | |
| **Pin#** | **Description** | | **Label** |
| **In Datasheet** | **In Schematic** |
| Reset | | | |
| 13 | ~MCLR | ~MCLR | ICSP\_~MCLR\_VPP\_TARGET |
| Clock | | | |
|  | T2CK | RPINR3 | Timer 2 External Clock |
|  | T3CK | RPINR3 | Timer 3 External Clock |
|  | T4CK | RPINR4 | Timer 4 External Clock |
|  | T5CK | RPINR4 | Timer 5 External Clock |
| IO Pins | | | |
| 17 | TMS/RPI16/RA0 | TMS/RA0 | TMS/RA0 |
| 20 | AN5/C1IN1+/VBUSON/VBUSST/RPI37/RB5 | C1INA/VBUSST/RB5 | VBUSON/C1IN+/AN5/CN7/RB5 |
| 21 | AN4/C1IN2-USBOEN/RPI36/RB4 | C1INB/USBOE/RB4 | USBOEN/C1IN-/AN4/CN6/RB4 |
| 22 | AN3/C2IN1+/VPIO/RPI35/RB3 | C21INA/RB3 | C2IN+/AN3/CNS/RB3 |
| 23 | AN2/C2IN2-/VMIO/RPI34/RB2 | C2INB/RB2 | C2IN-/AN2/CN4/RB2 |
| 24 | PGEC3/AN1/RPI33/RB1 | PGEC3/RB1 | PGC1/AN1/CN3/RB1 |
| 25 | PGED3/AN0/RPI32/RB0 | PGED3/RB0 | PGD1/AN0/CN2/RB0 |
| 26 | PGEC1/AN6/RPI38/RB6 | PGEC2/RB6 | ICSP\_PGEC\_TARGET |
| 27 | PGED1/AN7/RCV/RPI39/RB7 | PGED2/RB7 | ICSP\_PGED\_TARGET |
| 28 | VREF-/RA9 | VREF-/RA9 | PMPA7/VREF-/RA9 |
| 29 | VREF+/RA10 | VREF+/RA10 | PMPA6/VREF+/RA10 |
| 32 | AN8/PMA6/RPI40/RB8 | PMPA6/RB8 | C1OUT/AN8/RB8 |
| 33 | AN9/PMA7/RPI41/RB9 | PMPA7/RB9 | C2OUT/AN9/RB9 |
| 34 | AN10/CVREF/PMA13/RPI42/RPI42/RB11 | CVREF/PMPA13/RB10 | PMPA13/CVREF/AN10 |
| 35 | AN11/PMA12/RPI43/RB11 | PMPA12/RB11 | PMAPA12/AN11/RB11 |
| 38 | TCK/RPI17/RA1 | TCK/RA1 | TCK/RA1 |
| 41 | AN12/PMA11/RPI44/RB12 | PMPA11/RB12 | PMPA11/AN12/RB12 |
| 42 | AN12/PMA10/RPI45/RB13 | PMPA10/RB13 | PMPA10/AN13/RB13 |
| 43 | AN12/PMA1/RPI46/RB14 | PMPA1/RB14 | PMPA1/AN14/RB14 |
| 44 | AN12/PMA0/RPI47/RB15 | PMPA0/RB15 | PMPA0/AN15/OCFB/CN12 |
| 58 | ASCL2/RPI18/RA2 | ASCL2/RA2 | SCL2/RA2 |
| 59 | ASDA2/RPI19/RA3 | ASDA2/RA3 | SDA2/RA3 |
| 60 | TDI/RPI20/RA4 | TDI/RPIA4 | TDI/RA4 |
| 61 | TDO/RPI21/RA5 | TDO/RPIA5 | TDO/RA5 |
| 66 | RPI30/RA14 | RA14 | INT3/SCL1/RA14 |
| 67 | RPI31/RA15 | RA15 | INT4/SCA1/RA15 |
| 91 | AN22/RPI22/RA6 | RPIA6 | TRCLK/RA6 |
| 92 | AN23/RPI23/RA7 | RPIA7 | TRD3/RA7 |

Add rows to the example tables if necessary. Sort IO pins in ascending order.