Laboratory 3 tasks

*Alexander Lundqvist*

**For compiling and installing on chipKit**

make clean

make

make install TTYDEV=/dev/tty.usbserial-A503WFGA

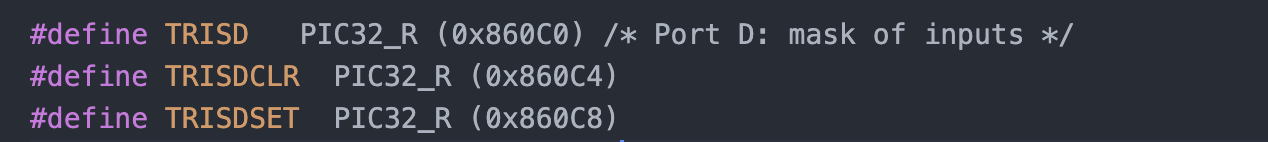
**Task 1**

*• Test pressing BTN3 and BTN2 at the same time. What happens? Why?*

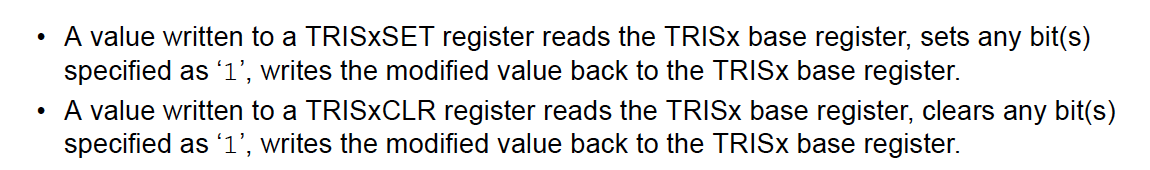
Both digit 3 and two gets updated value on screen. This is because the program actually executes all the if statements. We have bruteforce statements that says, for instance, that if we press 3&2 (011), 2 of the if-statements have this comparison.

*• Three device-registers for input/output control are TRISE, TRISESET, and TRISECLR.*

*Their functions are related. How? What are the differences?*



**12.2.8Set, Clear, and Invert I/O Port Registers**



TRISE: Determines if a pin should be an input (1) or an output (0)

TRISESET: Sets all the mapped pins to inputs

TRISECLR: Sets all the mapped pins to outputs

*• In the generated assembly code, in which MIPS register will the return values from*

*functions getbtns and getsw be placed in. You should be able to answer this question*

*without debugging the generated assembly code.*

$v0 and $v1

*• In this exercise, we explained which bits that should be used in Port D and Port E. How can*

*you find this information in the PIC32 and ChipKIT manuals? Be prepared to demonstrate*

*how to find this information in the manuals.*

*Advice: check the lecture slides from lecture 5 for ideas.*

Table 4-25 to 4-28 in the PIC32 Family Data Sheet

SW2 (pin 7), SW3 (pin 8) and SW4 (pin 35)

**Task 2**

*• When the time-out event-flag is a "1", how does your code reset it to "0"?*

By using the CLR register for IFS0 and clearing the 8th bit.

IFSCLR(0) = 0x0100

*• What would happen if the time-out event-flag was not reset to "0" by your code? Why?*

The timer on the display counts really fast. This indicates that the whole delay part is ignored.

It would never check for timeout events again. Timer register would never count up to period and thus there would be no desired delay.

*• Which device-register (or registers) must be written to define the time between time-out*

*events? Describe the function of that register (or of those registers).*

“PR is the period register. When a timer reaches the specified period, it rolls back to

0 and sets the TxIF bit in the IFS0 interrupt flag register. “

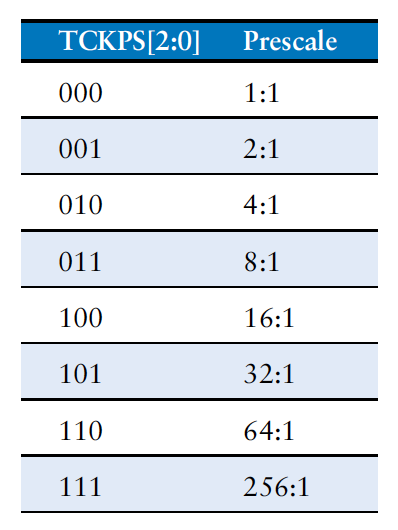
PRx? See p14 PIC32 Family Reference Manual Section 14 Timers.

The PRx register is the period register for the timer. It contains a value that is the length of a period.

*• If you press BTN3 quickly, does the time update reliably? Why, or why not? If not, would*

*that be easy to change? If so, how?*

Yes it does. Because ??

**PR2 choice of numbers!**

80 000 000 = 80MHz ( Clock frequency of chipkit)

256 = prescaling, can be changed but it was used in the lecture for the example with 1 sec.

10 = Because dividing by 1 gives 1 second, and 1000ms/10 = 100ms

**Task 3**

*• When the time-out event-flag is a "1", how does your code reset it to "0"?*

IFSCLR(0) = 0x100;

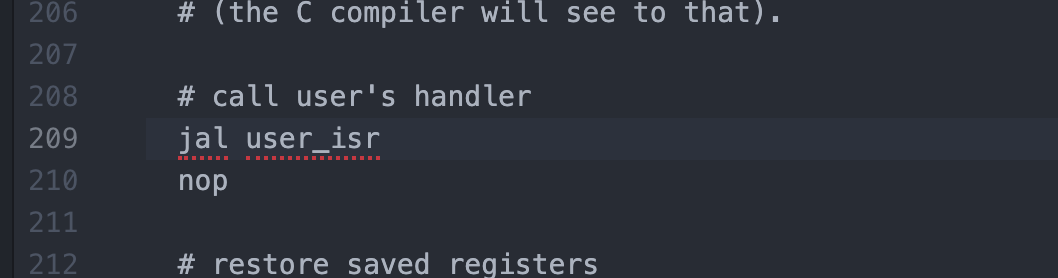
*• What would happen if the time-out event-flag was not reset to "0" by your code? Why?*

The timer on the display counts really fast. This indicates that the whole delay part is ignored.

It would never check for timeout events again. Timer register would never count up to period and thus there would be no desired delay.

*• From which part of the code is the function user\_isr called? Why is it called from there?*

Vectors.S



Because?????????

*• Why are registers saved before the call to user\_isr? Why are only some registers saved?*

The registers that are saved are of the non-preserved type.

Since we have a function \_isr\_trampoline (caller) that calls user\_isr (callee), it is the callers responsibility, according to to general practice, to save the non-preserved registers in case it wants to use them after the function call.

*“Registers $1 through $15, $24, and $25 can be changed by any C function, or any*

*assembler subroutine. Register $ra will be changed by the call to the C function, i.e.,*

*the jal instruction. Saving these registers is necessary before calling the C function*

*user\_isr.”*

*• Which device-register (or registers), and which processor-register (or registers) must be*

*written to enable interrupts from the timer? Describe the functions of the relevant registers.*

IEC0 – Interrupt register with interrupt enable control, bit 8 is T2IE

IPC2 - Priority