

EE 308 – LAB 7

HC12 Timer Input Capture and Output Compare

Last week you wrote programs using the HC12 Timer Overflow Interrupt and Real Time Interrupt. This week you will work with the Timer Input Capture and Timer Output Compare functions.

1. Start with the following program, which is just a do-nothing infinite loop:

```
#include "hc12b32.h"

#define TRUE 1

main()
{
    DDRA = 0xff;          /* Make all bits of PORTA output */
    PORTA = 0x00;
    while (TRUE)
    {
        _asm("wai");
    }
}
```

2. Connect one of your debounced switches to Input Capture 2 (PORTT, Bit 2), as shown below:

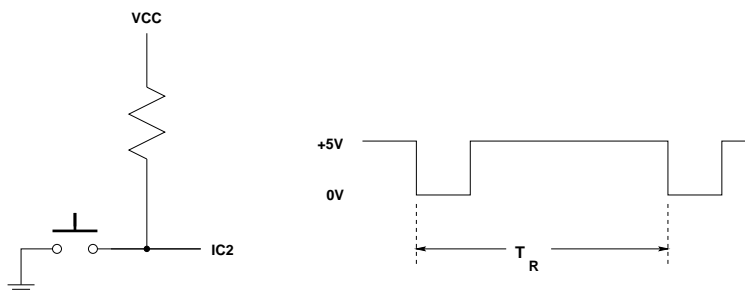


Figure 1: Circuit to measure speed of button pushing.

The right part of Figure 1 is what the signal to IC2 will look like if you push the pushbutton twice.

3. Add to your program code to measure the time T_R between the two falling edges of the signal in Figure 1. Set the prescaler so you can measure time differences up to 250 ms.

It is inconvenient to have to stop the HC12 and look at a memory location to determine what your result was. D-Bug12 has built in routines to write information to the terminal. This is documented in the application note **Using the Callable Routines in D-Bug12** included with your HC12 board. In your program, T_R should be a sixteen-bit unsigned number. D-Bug12 provides a C-like `printf()` function you can use to print information to the screen. The information below shows some examples of how to do this.

To use the D-Bug12 routines you will have to include the header file `DBug12.h`. This can be downloaded from the EE 308 homepage.

Here is a program to print `hello, world` to the terminal:

```
#include "DBug12.h"

main()
{
    DBug12FNP->printf("hello, world\n\r");
}
```

Here is a program to print an unsigned number to the screen in both hexadecimal and decimal:

```
#include "DBug12.h"

main()
{
    unsigned int x;

    x = 0xf000;
    DBug12FNP->printf("Hex:  %x, Decimal:  %u\n\r",x,x);
}
```

Add to your program a `printf()` function to print out the result – the number of timer ticks between pushes of the button. (Do not try to calculate the actual time as a floating point number. When you use floating point numbers the programs will be too large to load into the HC12 without using expanded memory.) You should write your program as an infinite loop so that after pressing the button twice your program will print out the result, then go wait for the next two presses.

4. Test your program on your EVBU. See how fast you can push the switch twice.
5. Add an Output Compare function on Bit 3 of PORTT to generate a 1 kHz square wave. Use a logic probe to verify that Bit 3 of PORTT is toggling.
6. Add an Output Compare function on Bit 4 of PORTT to generate the *friend* signal described in Problem 1 on Page 291 of Pack and Barrett. This signal consists of five pulses which are high for 100 μ s and low for 100 μ s, followed by a 1000 μ s low signal. This signal then repeats.

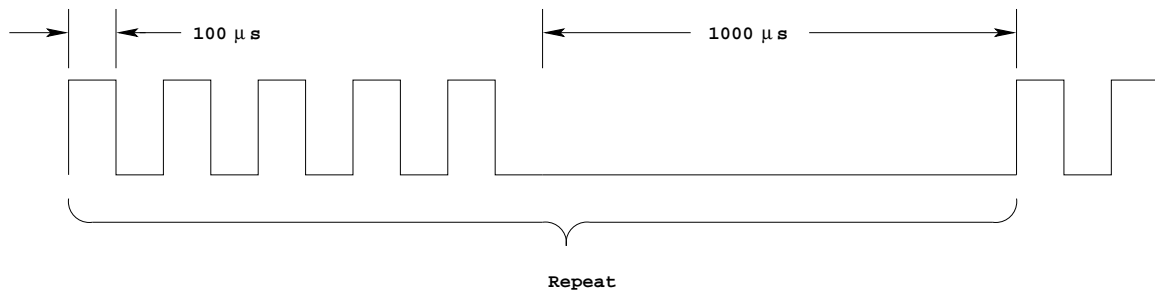


Figure 2: Friend signal for Part 6.

7. Connect your square wave and friend signals to a logic analyzer. Verify that the square wave has a 1 kHz frequency and a 50% duty cycle, and that the friend signal looks like the Figure 2.

Note: We are replacing the logic analyzers in the digital lab. There is no longer a logic analyzer at each station. There are now five stations with dedicated logic analyzers. To use one of the logic analyzers you will need to take your wiring kit from Digital Lab and connect wires to ground, Bit 3 of PORTT, and Bit 4 of PORTT. You will take your HC12 board to one of the logic analyzer stations, connect the ground and two signals to the logic analyzer, and capture a waveform. The software for the new logic analyzers is a little different than the software for the old analyzers. Your lab instructor will help you with the logic analyzer software and hardware.