Lab 10 - Input Compare Timer Functions.

Instructor: Fred Etcheverry

Review **Chapter 8**, 68HC11 Timer Functions, **Sec. 8.4 and 8.5**, Input-Capture Functions, ***MC68HC11 An Introduction Software and Hardware Interfacing*** by Han-Way Huang. All ***CME11E9-EVBU Development Board Instruction*** Reference: ***Manual***. Download form www.axman.com.

Equipment: ***CME-11E9-EVBU Development Board***. Purchased from Axiom Manufacturing Inc. www.axman.com.

Parts: 2-100Ω, 2-1KΩ, 5-10KΩ 1/4W resistors, 1-0.1μF Disk Cap, 1-LM358 Op-Amp., 1-LM311 Comparator, 1-Electret Condenser Microphone Cartridge -44dB Sensitivity or higher (WM-54PC), and #22 solid hook-up wire.

Software: AxIDE v. 2. comes with Development Board or can be downloaded.

1. Make sure that the MODA Jumper in moved from OFF to ON position. See page 16 of the Axiom Instruction Manual. This will change the ***Mode of Operation*** from ***Normal Expanded*** to ***Normal Single Chip Operation.***

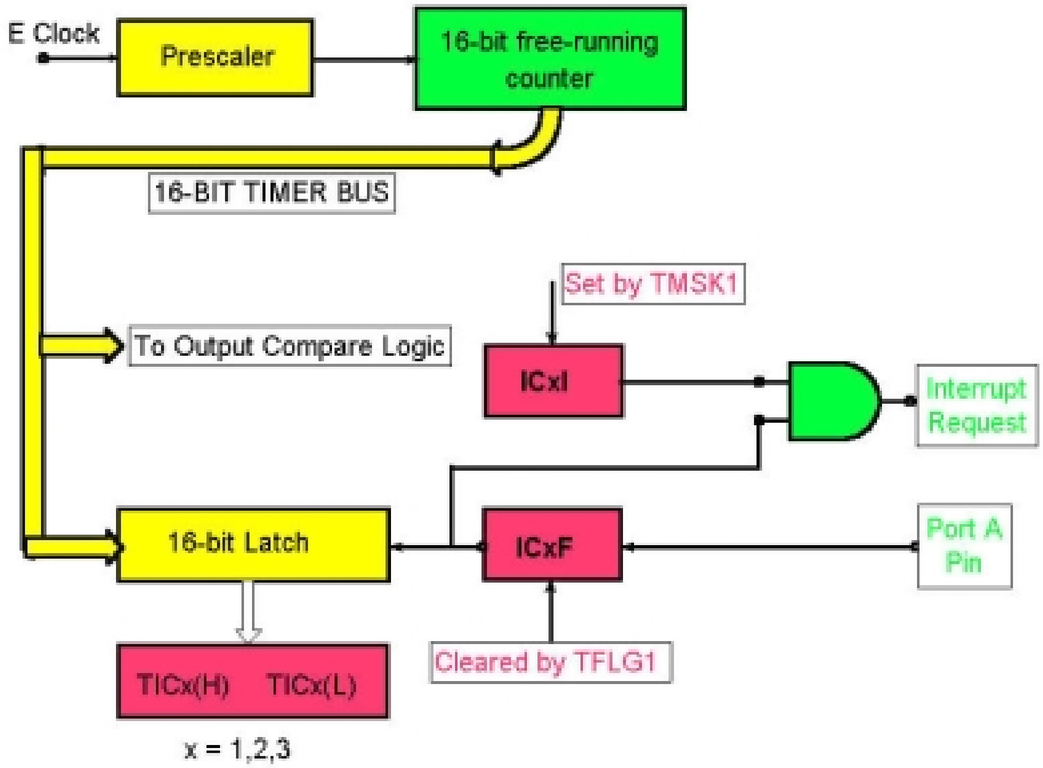
2. Remove the MAN6740 Dual 7-Segment LED Display connected from the previous lab.

Input Compare Timer Function

The 68HC11 ***input compare timer function*** can record the time that a pulse arrives. The program can configure the input compare to respond to a pulse's rising edge, or falling edge. A flag is set when any pulse is detected by any input compare and can be reset by the program. Masks bits can be set to enable any of three ***timer input compare*** interrupts.

Timer Input Compare (TIC)

When the proper pulse edge arrives at the proper pin on Port A, a ***Input Compare Flag*** is set and the output of the 16-BIT FREE-RUNNING COUNTER is fed to a 16-BIT LATCH. (See Figure 10.1.) This latch stores the value of the counter and feeds a TIMER INPUT COMPARE (TIC). Each TIC is a 16-bit register that can be read under program control. Each Input Compare Flag drives an AND gate which is also fed by an ***Input Compare Interrupt*** mask.



	7	6	5	4	3	2	1	0
TMSK1 Enable = 1 Interrupt Masks	OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I
TFLG1 Clear = 1 Compare Flags	OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I

Figure 10.1: 68HC11 Input-Capture Diagram. The 68HC11 E Clock is fed to the Prescaler. The EVBU E Clock is 2MHz. The Prescaler is set by BUFFALO to divide by one. Thus the output of the Prescaler is also 2MHz. The output of the Prescaler is fed to a 16-bit free-running counter. This counter drives a 16-bit timer bus, which drive three 16-bit latches. A pulse with the proper edge arriving at Port A pins PA/ICx x=1,2,3 sets Input Compare Flag ICxF which causes a 16-BIT LATCH to store the output of the COUNTER. The output of the LATCH feeds a Timer Input Compare TICx which can be read by the program. At the same time that the Input Compare Flag feeds the 16-bit Latch, it drives an input to an AND gate. The Input Compare Interrupt bit drives the other input of the AND gate. The output of the AND gate is high (indicating an interrupt) only when the interrupt is set and the flag is set.

Software Interrupt

The 68HC11 software interrupt is initiated by the SWI command. This command causes the program counter to go to a jump table. BUFFALO is programmed such that the service routine for this interrupt returns to BAFFALO and displays the contents of the CPU registers. The SWI command is a powerful troubleshooting tool since it allows us to break the program at any point and observe program parameters.

Flag Driven Program

Before the advent of sophisticated interrupts, ***flag driven*** programs were prominent. Such a program loops until it finds a device flag set. It then services the device, resets the flag and resumes. In this lab, we will use a flag driven program with the SWI command to develop and troubleshoot a program using input compare interrupts. The input compare function changes the state of an internal register (TIC) in response to an input pulse. A flag driven program with an SWI command lets us test this TIC by looping on an interrupt flag until it is ready (the selected pulse edge has arrived) and then loading the contents of the TIC register into a CPU register.

Constructing the Interface

3. Connect the following circuit.

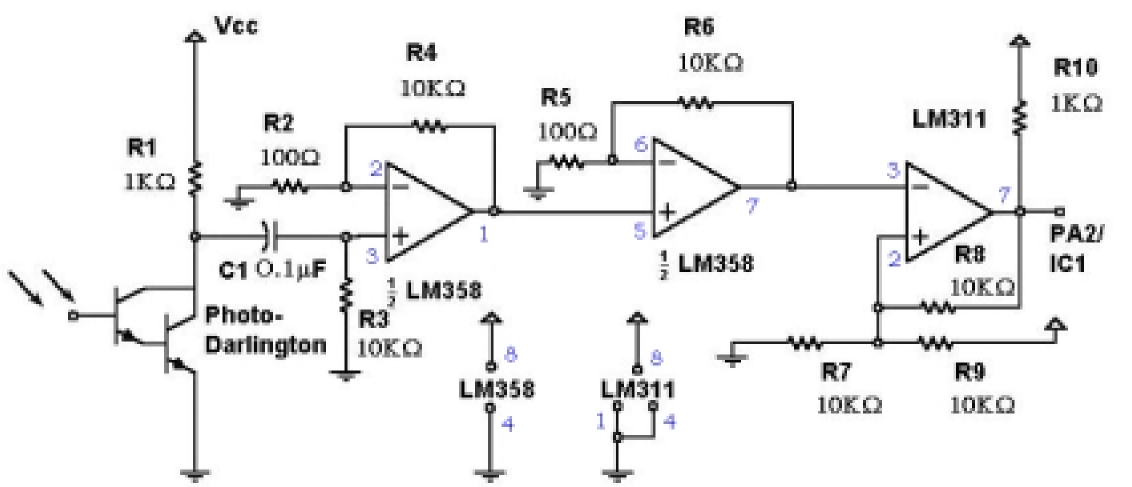


Figure 10.2 Photo Detector Interface Schematic. A Photo-Darlington transistor which is the detector side of an H22A1 Slotted Optical Switch drives an amplifier and a comparator. The amplifier is constructed from the two op-amps found on the LM358. The comparator is an LM311. The power supply connections (Vcc and ground) to the LM358 and LM311 are shown.

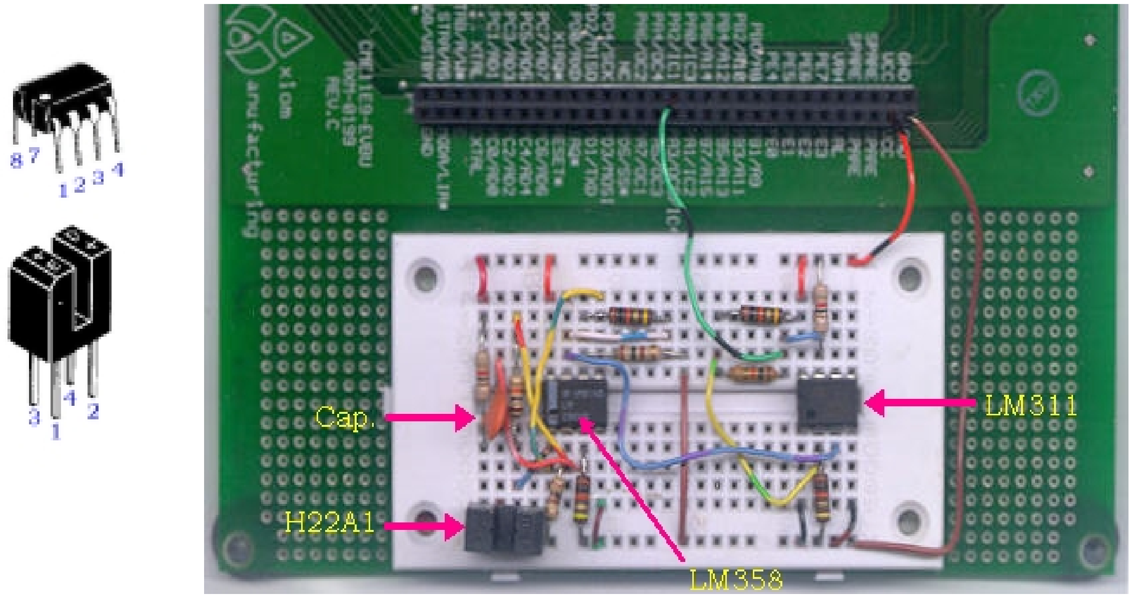


Figure 10.3 Photo Detector Pictorial. The above interface is constructed on CME11E9-EVBU Development Board. Both the LM358 and LM311 are 8-pin DIPS. The H22A1 Slotted Optical Switch pin-out is shown. In this lab we only need the detector section of the H22A1 marked on top the device as D and + . Pin 2 is the collector and 4 is the emitter. (Photo of CME11E9-EVBU with permission of Axiom Manufacturing)

How the interface works

The light from an incandescent lamp fluctuates at a frequency twice that of the line frequency (60Hz or 50Hz) since the lamp pulses once during the positive swing of each cycle and again during the negative swing. When an incandescent lamp is placed near the photo-darlington, a signal at twice the line frequency appears at the collector. This signal is capacitively coupled into an amplifier consisting of two op-amps configured as non-inverting amplifiers. The output of these amplifiers is then fed to a comparator. The LM311 is designed to drive digital inputs. It shapes the waveform giving it fast changing edges. Since the output of the LM311 is open collector, R10 is required as a pull-up resistor. Resistors R7, R8, and R9 form a network that gives the LM311 hysteresis. That is, the output of the LM311 goes low only after its input voltage goes above the **upper threshold voltage** $\mathbf{V_{UTH}}$, and its output only goes high after the input voltage goes below **the lower threshold voltage,** $\mathbf{V_{LTH}}$**.** There is a difference between these two thresholds called the **hyteresis**; it provides a **noise margin** for the input.

The 68HC11 edge detection circuits, like any digital input circuitry, can only respond to rapidly changing edges. Recall that frequency and pulse width are reciprocals. For example, a 60Hz line frequency will produce a flicker rate twice 60Hz or 120Hz. Thus 1/120Hz = 8.33milliseconds.

This circuit responds even better to fluorescent lamps. Many modern fluorescent lamps, however, incorporate high frequency inverters. (About 25KHz) This is done to remove the annoying flicker observed in older type lamps. This interface provides a way to measure the frequency of any fluorescent. Remember that the flicker frequency is twice that of the lamp's supply.

In this lab we are only using the detector side of the H22A1 Slotted Optical Switch. In the next lab we will use this device as an optical switch. Its use in this lab illustrates the problem of noise from external light sources.

Capturing a pulse edge with Input Compare

3. Assemble the following program.

```
*This program prepares the TIMER INPUT COMPARE
*register 1 to detect a rising edge. It sets
*the Y index register to point to the top
*of the stack. This program reads TIC1
*edge 1 and edge 2 and puts the difference
*on the top of stack. The program then jumps to BUFFALO.
*The register are displayed. The delay between the
*edges is displayed in registers A & B.
REGBAS   EQU      $1000
TFLG1    EQU      $23      ;TIMER FLAGS
TMSK1    EQU      $22      ;TIMER INTERRUPT MASK
TIC1     EQU      $10      ;TIMER INPUT COMPARE
TCTL2    EQU      $21      ;TIMER CONTROL EDGES
R_EDGE   EQU      $10      ;SET IC1 RISING EDGE
IC1F     EQU      $04      ;CLEAR IC1 FLAG SET MASK
         ORG      $B600
START    LDS      #$01FF
         DES               ;PREPARE
         DES               ;Y = TOP OF STACK.
         TSY
         LDX      #REGBAS ;SET X TO POINT TO BASE REGISTER.
         LDAA     #R_EDGE ;IC1 RISING FLAG
         STAA     TCTL2,X ;TCTL2 TIMER CONTROL
         LDAA     #IC1F
         STAA     TFLG1,X ;SET IC1 INTERRUPT FLAG
TEST     LDAA     #IC1F
         ANDA     TFLG1,X ;TEST TFLG1 FLAG
         BEQ      TEST
         LDD      TIC1,X  ;READ TIMER INPUT COMPARE 1.
         STD      0,Y      ;GET FIRST EDGE.
         LDAA     #IC1F
         STAA     TFLG1,X ;SET IC1 INTERRUPT FLAG.
TEST2    LDAA     #IC1F
         ANDA     TFLG1,X ;TEST TFLG1 FLAG
         BEQ      TEST2
         LDD      TIC1,X  ;READ TIMER INPUT COMPARE 1.
         SUBD     0,Y      ;SUB EDGE1 FOR EDGE2 TIME.
         SWI               ;RETURN TO BUFFALO.
```

Before you type "LOAD T," remember you must change the baud rate of the upload.

4. Place a 100-watt lamp or greater about 25cm (10-inches) above the optical switch. Run the program.

Notice that the program should only return to BUFFALO if pulses are input to the 68HC11. If the interface is not producing pulses then the program will not fall through loops (TEST and TEST2).

If you have access to an oscilloscope, you can troubleshoot the interface by checking for pulses. Start at PA2/IC1. If there are no pulses there of about 0 to 5-volts, then work back to the op-amps and check for pulses at their outputs.

You can also troubleshoot the interface by introducing pulses. Run the program. Start at the port by connected a wire to PA2/IC1. Touch the other end of the wire to V_{CC} and then ground; repeat this several times. This will introduce pulses into the 68HC11 and the program should fall through the loops, and return to BUFFALO. If this works keep moving back connecting the wire to the input of each op-amp and continue pulsing until the defective stage is found.

Analyze Program

After the program is run, which registers contain the pulse width?

5. Run the program ten times and record the pulse width in microseconds.

Are the values recorded approximately equal to the calculated value of the pulse width?

6. Examine the ten recorded values of the pulse width. Find the maximum deviation form the calculated value. Max deviation = ____________.

7. Repeat steps 5 several times experimenting with lamp position and any source of external noise such as a computer monitor or fluorescent lamp.

How is the deviation of the pulse width effected?

8. Temporarily remove R8. This will remove the hyteresis from the comparator.

9. Repeat step 7.

How does removing the hyteresis from the comparator affect the deviation of the pulse width?

Calculating the Hysteresis of the Comparator

The comparator circuit shown in Figure 10.4. The output of the LM311 comparator goes high (V_{CC}) if the voltage fed to the inverting input (-) if less than the voltage fed to the non-inverting input (+). The output of this comparator goes low (ground) if the voltage fed to the inverting input is greater than the voltage fed to the non-inverting input.

To calculate the hysteresis assume the comparator is a switch that connects the output to ground when it goes low and opens to let the pull-up resistor pull high. Also assume that the non-inverting input does not draw any current. The voltage of the upper threshold V_{UTH} is thus the voltage at the non-inverting input when the output is high, and the voltage of the lower threshold V_{LTH} is the voltage at the non-inverting input when the output is low. The difference of these to threshold voltages is the voltage of the hysteresis V_H.

10. Calculate V_{UTH}, V_{LTH}, and V_H.

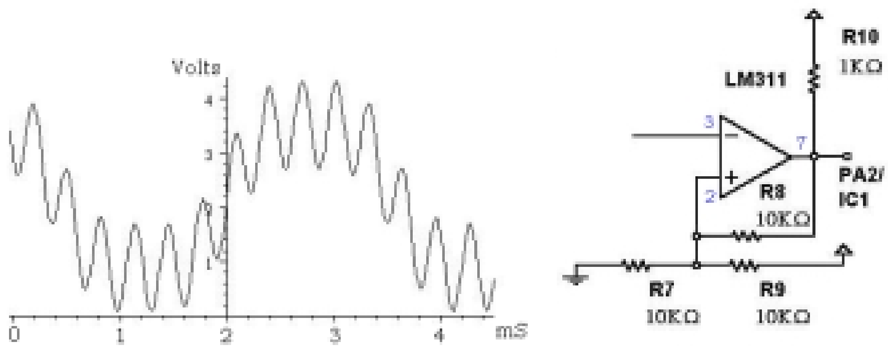


Figure 10.4 Waveform of signal with noise and a comparator circuit designed to eliminate some of the effects of such noise.

11. Assume that the comparator shown in Figure 10.4 is fed a signal represented by the waveform to its left. Draw the output waveform of the comparator with hyteresis and without (R8 removed).

12. Modify the above program to convert the pulse width to millisecond. Load the whole number of milliseconds into the A-Register and the tenths into the B-Register. Here is a suggested strategy:

1. Create two extra locations on the stack for temporary storage.
2. Use the IDIV instruction to find the whole number of milliseconds.
3. Save the remainder.
4. Use the IDIV instruction to find the number of tenths of a millisecond in the remainder.

13. Replace the Photo-Darington in the above circuit with an electret (condenser) microphone cartridge -44db sensitivity or better.

14. Run the program. Play a musical instrument, sing, use a tuning fork or cookie to produce a tone of a known frequency.

The note A above middle C is 440Hz. Each octave higher doubles in frequency and each octave lower halves in frequency. Blowing across the opening of a tube closed at the other end such as a bottle will produce a tone with a pulse width P_W is given by:

$$P_W = \frac{4L}{S} \qquad (10.1)$$

where L is the length of the tube in meters and S is the speed of sound (about 330 Meters /Seconds).

A tube open at both ends will produce a note with a pulse width given by:

$$P_W = \frac{2L}{S} \qquad (10.2)$$

How does the pulse width measured by the program agree with the calculated values?

[NOTE: If the calculated values are integer multiples of the measured values then the program is measuring harmonics. One solution to emphasize the fundamental (lowest frequency) would be to increase either C1 or R3.]

If you have access to a function or signal generator the program can measure the pulse width of its output. First make sure that the generator ground is connected to the EVBU ground. If the generator has a TTL (digital output), connect this output through a 1KΩ resistor to PA2/IC1 (remove the wire connecting the 311 to PA2/IC1). If the generator does not have a TTL output then connect it through a 1KΩ resistor to pin 5 of the non-inverting input of the LM358 op-amp (remove the wire from pin 1 to pin 5).

15. Run the program with a signal source (sound or genrator). Record the results of several pulse widths.

How do they check with calculation?

This interface has converted the 68HC11 into a pulse width meter. By using arithmetic instructions we could make this interface function as a frequency meter. [By dividing the pulse width into 1.] In the next lab we will construct a **frequency counter**.