Debugging Interrupts

- Debugging systems with many interrupts can be difficult.
- The primary need is for visibility into what is happening.
- By using a primitive 3-bit DAC and some debug code we can determine how often interrupts occur and how long it takes for them to run.
- By commenting out some code you can also tell if a interrupt is hung in an endless loop.

Debugging Interrupts

- The DAC circuit is shown below. This example is for use on the ATMega128 board supplied by OSU.

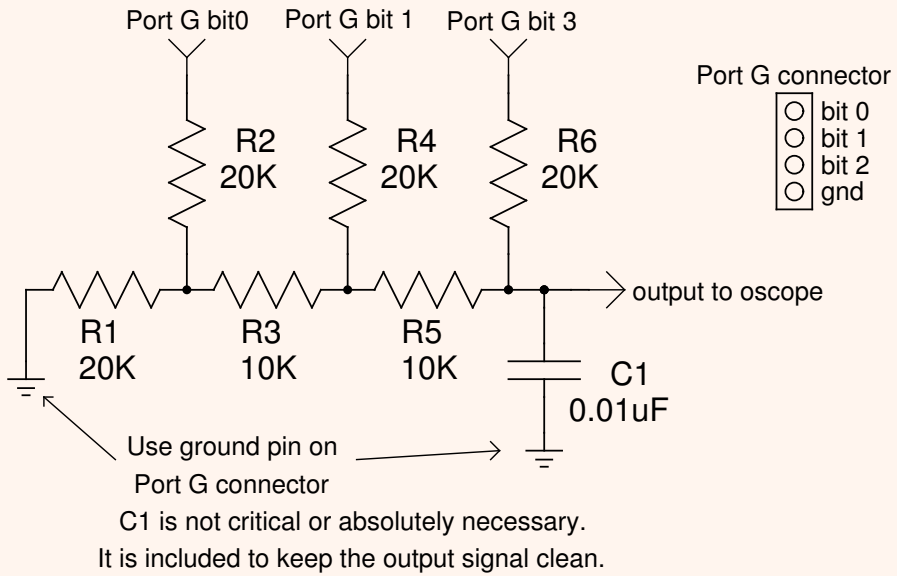

Figure 1: R2R Network DAC

- The output voltage of this DAC is:

$$V_{out} = V_{ref} * \frac{bit\_value}{2^{number\_of\_bits}}$$

Debugging Interrupts

- Since we have 3 bits and our reference voltage is 5 volts, the output voltages we can obtain with different bit values is:

  $V_{out} = 5 * \frac{bit\_value}{8}$

Bit value	Output Voltage
000	0V
001	0.625V
010	1.25
011	1.875
100	2.5
101	3.125
110	3.75
11l	4.375

Debugging Interrupts

- In our code, we associate a bit value and thus a voltage for each interrupt.
- When we are executing in `main()` the output voltage is zero.

```
//3-bit DAC binary values
#define TCNT0_ISR    0x01
#define TCNT1_ISR    0x02
#define TCNT3_ISR    0x03
#define ADC_ISR      0x04
#define TWI_ISR      0x05
#define USART0_ISR   0x06
#define NOT_IN_ISR   0xF8  //reset lower three bits to zero
}
```

Debugging Interrupts

- We get distinct voltages for each interrupt.

Bit value	Output Voltage	Running Code
000	0V	main()
001	0.625V	TCNT0 ISR
010	1.25	TCNT1 ISR
011	1.875	TCNT3 ISR
100	2.5	ADC ISR
101	3.125	TWI ISR
110	3.75	USART0 ISR
11l	4.375	unused

Debugging Interrupts

- We use one variable to determine if the interrupt debug code is used.

```
//define to show interrupts via 3-bit DAC
#define SHOW_INTERRUPTS
```

- The beginning and end of each ISR is bracketed with debug statements.

```
#ifdef SHOW_INTERRUPTS
  PORTG |= (USART0_ISR); //set interrupt value to this ISR
#endif

//ISR code goes here

#ifdef SHOW_INTERRUPTS
  PORTG &= NOT_IN_ISR;  //set interrupt value back to zero
#endif
```

Debugging Interrupts

- For example, the ADC ISR looks like this.

```
ISR(ADC_vect){
#ifdef SHOW_INTERRUPTS
  PORTG |= (USART0_ISR); //set interrupt value to this ISR
#endif

OCR2=ADCH;           //update pwm dimmer
adc_flag=ADC_DONE; //set done flag

#ifdef SHOW_INTERRUPTS
  PORTG &= NOT_IN_ISR;  //set interrupt value back to zero
#endif
}//ADC_vect_ISR
```

Debugging Interrupts

- Five different interrupts, their periods and duration seen here. Plenty of time for `main()` and other interrupts to run. This is a good picture.

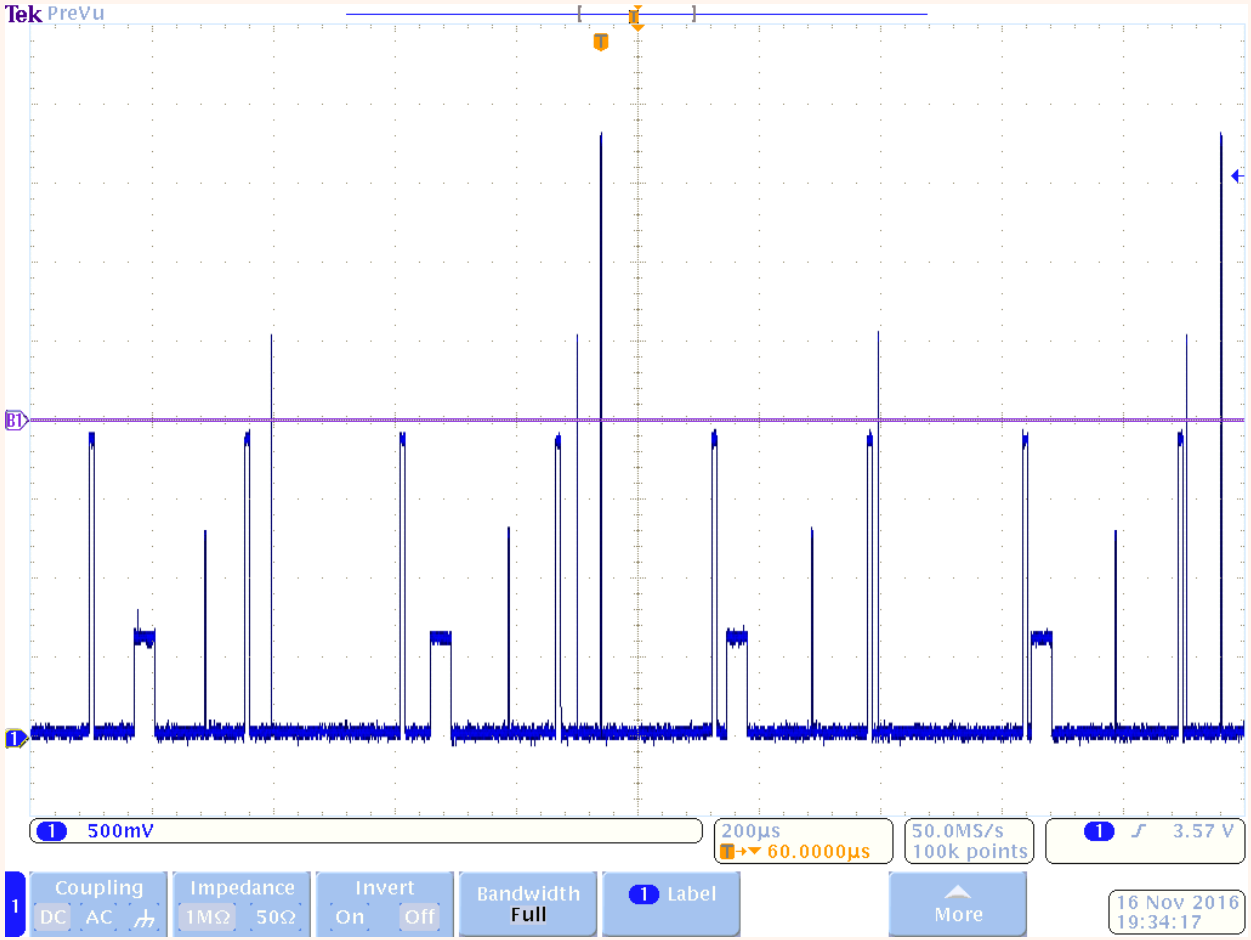

Figure 2: Big Picture of All Interrups in a System

Debugging Interrupts

- You can determine relative or absolute ISR run times if you zoom in.

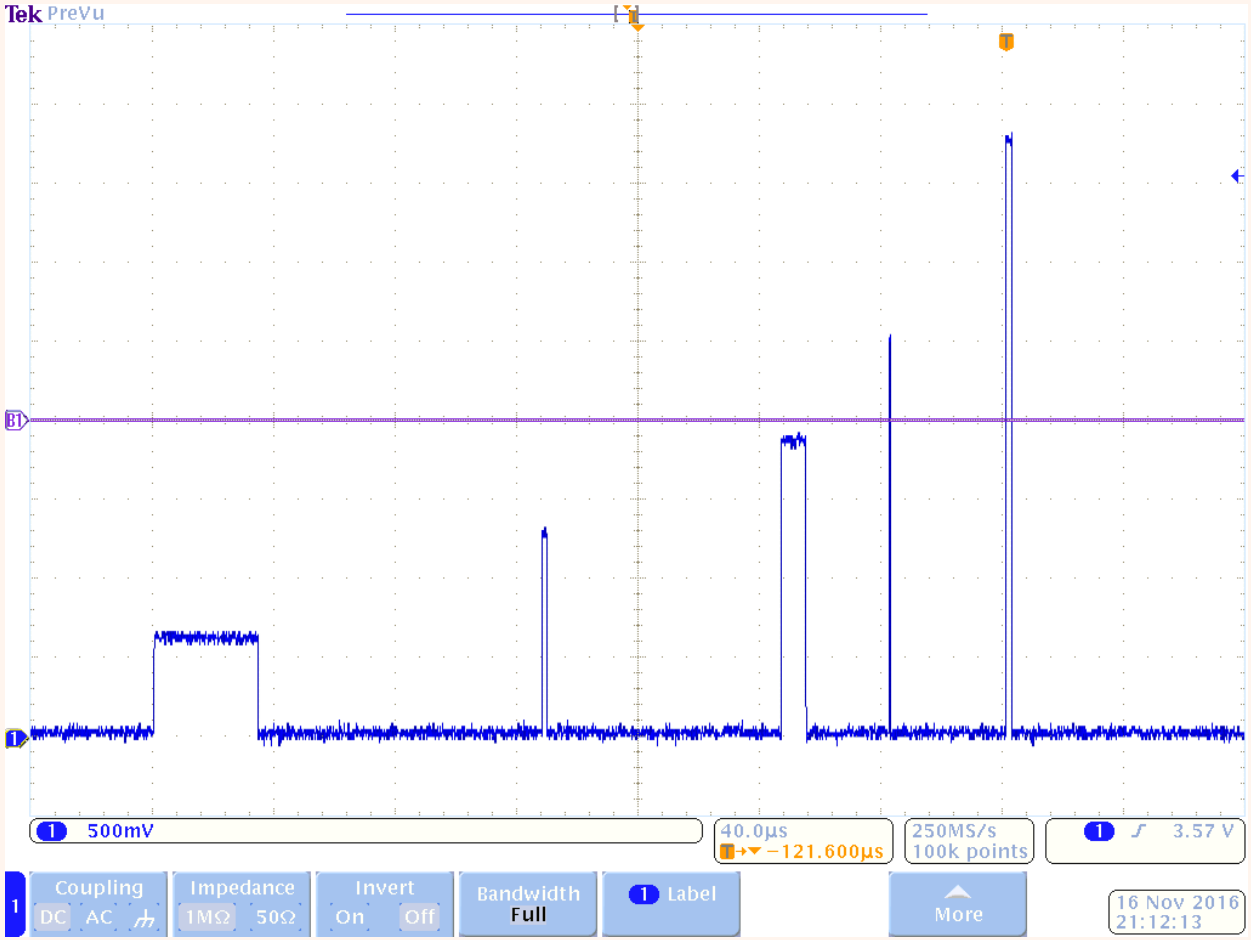

Figure 3: Relative Timing of Interrupts Shown

Debugging Interrupts

- ISR for TCNT0 (0.625V) ran 34.6uS, USART0 ISR (3.77V) ran 2uS.

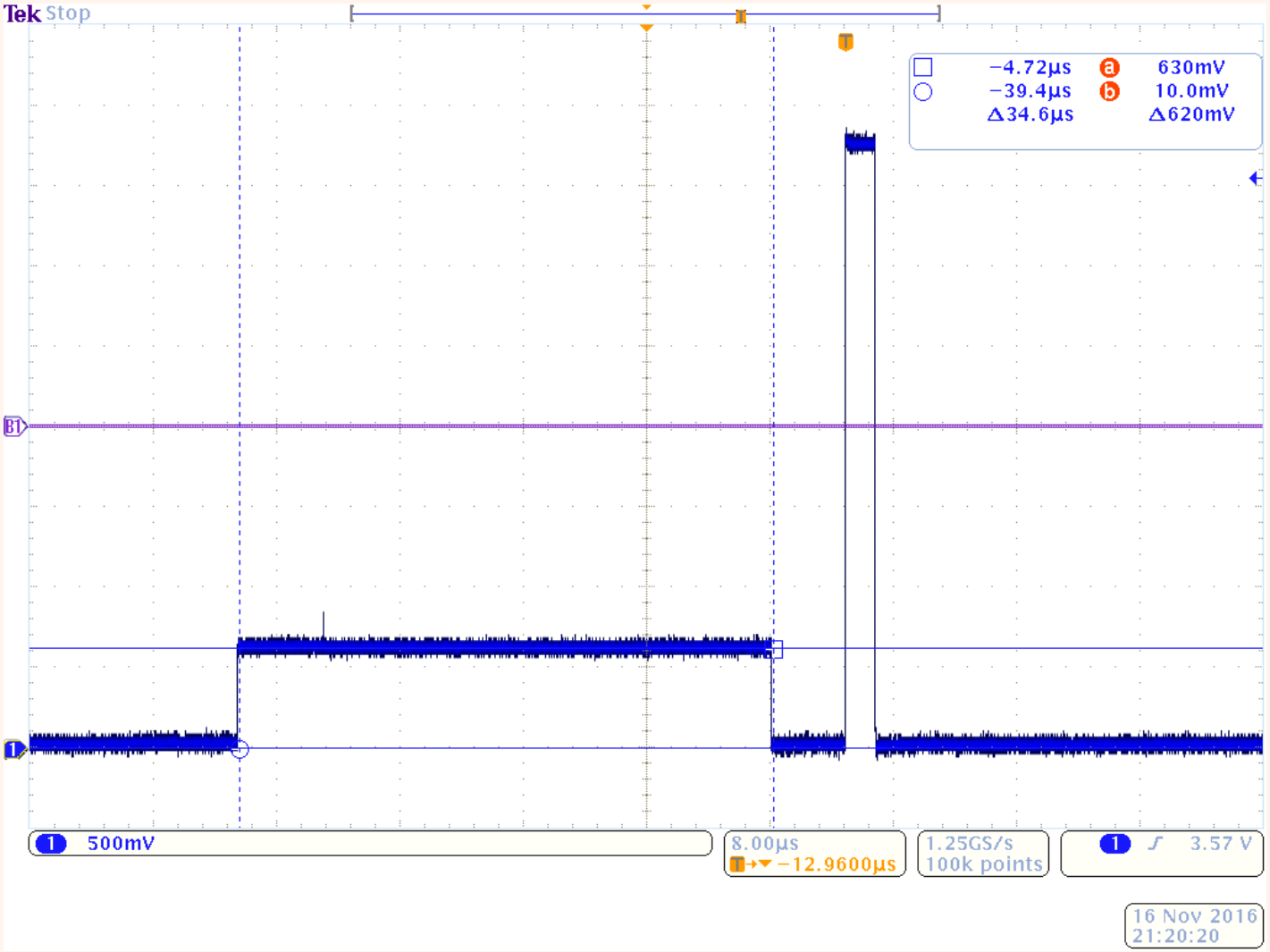

Figure 4: Interrupt Timing