Application Note AN2177

Ultra-WideBand RADAR Test Platform

Author: Peter A. Stephens

Associated Project: Yes
Associated Part Family: CY8C27xxx
PSoC Designer Version: 4.0

Summary

This Application Note covers a PSoC-based Ultra-WideBand RADAR test and demonstration platform. It uses a Cypress MicroSystems CY8C27443 PSoC as a controller and to provide many of the required analog functions. This helps to make it a compact Ultra-WideBand RADAR system. The intent of the system is to allow the user to evaluate aspects of Ultra-WideBand RADAR such as the effect of pulse rate, number of pulses, transmit power level, etc.

Introduction

The term Ultra WideBand (UWB) brings visions of an extremely complex system. A system that can only work if everything is just perfect. Some of this vision is true. Some of it is false. A UWB RADAR system can be much simpler than a UWB communications system. Aspects of a UWB RADAR system do have to be perfect, but the system does not have to be overly complex.

RADAR is an acronym for, “Radio Detection and Ranging.” RADAR uses radio waves to detect and determine the range of a target item. It is described in many places as being similar to a sound echo in a canyon. The distance between the source (your shout) and the target (opposite canyon wall) can be estimated because the farther away the canyon wall is, the longer the echo takes to return.

A German engineer patented the first system using RADAR principals in 1904. Britain was one of the first countries to find practical use for it. This practical system was developed around 1934 for the military to allow detection of aircraft so that fighters could be vectored to intercept. The patent for this system describes both continuous wave and pulsed RADAR systems.

Pulsed RADAR Systems

A pulsed RADAR system sends out a pulse containing a signal of a known frequency. The time that it takes for the echo to return is used to compute the distance from the RADAR site to the target. A pulsed RADAR system can detect moving and stationary objects. Pulsed RADAR systems can be constructed with a single antenna where the antenna is switched between transmit and receive signal paths. The pulse signals are sometimes frequency or amplitude modulated to achieve better resolution on range measurements.

Some common uses for pulsed wave systems include:

- Liquid level detection
- Air control
- Targeting

Common parameters for the pulsed RADAR systems include: pulse width, pulse shape, pulse repetition frequency, average power, and carrier frequency. Wider pulse widths give a longer range. However, if the pulse width is too large, closer targets will reflect the leading edge before the system has finished limiting the minimum detection range. In a similar manner, the pulse shape can affect the accuracy, resolution, and range extents. If the rising edge of the pulse is skewed, the timing accuracy of the return pulse will be changed. A change in the trailing edge would harm the resolution by widening the pulse.

Distortion of the leading and trailing edges also makes the pulse wider, worsening the minimum range. High pulse repetition frequencies allow the system to detect targets at shorter ranges since the system is less likely to be in transmit mode when the pulse returns from its shorter trip. These same higher rates also increase the detection probability by allowing the system to do more averaging.

Continuous Wave RADAR Systems

Continuous wave RADAR sends out a signal of a known frequency. Echoes from targets shift away from the transmitted signal's frequency (Doppler effect). A disadvantage of continuous wave systems is they can only detect moving targets because targets at rest do not cause a Doppler shift. A version of this type uses an amplitude or frequency modulation technique to overcome this problem where a known frequency or amplitude deviation is used to modulate the carrier over time. This is then used to calculate the distance. This type of RADAR can be bulky because it requires 2 antennas, but it is more difficult to jam than a pulsed system. Because continuous wave RADAR does not pulse, a basic system can be simpler in design than pulsed systems, use a lower peak power, and have no absolute minimum or maximum ranges for most detection only uses. Some common uses for continuous wave systems include:

- Early warning systems
- Radio altimeters
- Land mine detection

Both continuous wave and pulsed systems benefit by adding more power to get better range. Since continuous wave systems have a lower peak power for a given average, they gain an advantage. Both systems also have to make tradeoffs such as carrier frequencies. A higher frequency allows smaller equipment, but lower frequencies better penetrate the atmosphere.

Ultra-Wideband RADAR Systems

Ultra-Wideband RADAR systems are similar to pulsed systems. Unlike pulsed systems that concentrate their energy at a small range of frequencies, UWB systems try their best to spread this energy to an extremely wide band of frequencies. They do this by approximating a unit impulse.

By its very nature, the UWB system has a built in range gate. Since the system is pulsed and only accepts inputs at the time (distance) it is looking at, it rejects inputs at the wrong time.

Since the pulse is very narrow and the time between pulses is large, the system has a higher peak power demand, but can be made to have a lower average power than other RADAR systems. UWB systems are usually used for shorter ranges because they are power limited by the FCC. Because of their low average power and potential low cost they may some day be ubiquitous. The basic idea of a UWB-based RADAR system is to send out a pulse and then look for a return 'echo' at the exact time it should come back for the distance you desire. You do this by triggering the receiver with a delayed version of the transmitted pulse.

$$t_{echo} = 2 \times \frac{dist}{c}$$

Figure 1. Echo Time Equation

The equation above (Figure 1) gives the round trip time for a pulse where 'dist' is the distance and 'c' is the speed of light. For example, a short (~3 nS) pulse should be able to go one meter and return in 6.7 nS.

Some possible uses for a UWB system gleaned from Tom McEwan's patent [1] include:

- Security Systems
- Concealed switches in junction boxes to turn on lights
- Doorways monitored by UWB sensors to open them and turn lights on
- UWB sensors may also help the handicapped
- Appliances can be activated for convenience and safety
- Life Detector For Disaster Work
- Heart beat and arterial pulse detection
- Automotive Collision Sensor

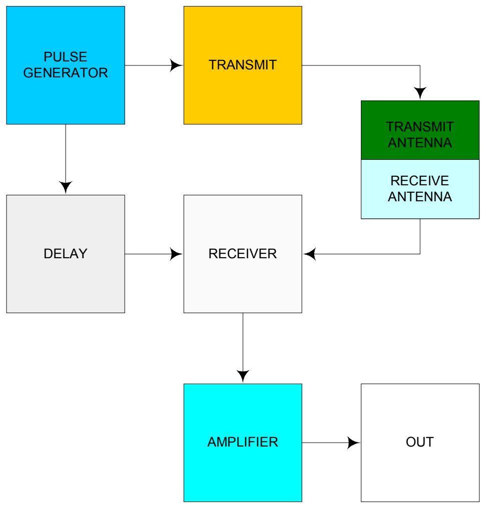


Figure 2. UWB RADAR Block Diagram

The diagram in Figure 2 should suffice for a quick discussion on a UWB RADAR system. The PULSE GENERATOR makes a trigger for a pulse and sends that to TRANSMIT and DELAY at the same time. TRANSMIT turns the trigger into a narrow pulse and sends it to ANTENNA, which emits the signal. For our discussion, DELAY causes a 6.7 nS delay in the trigger pulse and sends it to RECEIVER.

Assuming a target was in the appropriate place (1 meter), the ANTENNA then receives the signal echo and sends it to the RECEIVER at exactly the same time as DELAY sends the trigger to the RECEIVER telling it to sample the ANTENNA. This sample is sent to the AMPLIFIER and to the OUTPUT. With some exceptions, this is what is described in a UWB RADAR patent [1].

Design

Here is where we get to a fun part because this is the where we start partitioning the system-essential requirements to hardware and software. We also add the missing parts that were not part of the essential aspects of the system, but are necessary to make the system usable (sounds like a conundrum, but helps to separate what needs to be done from how it gets done[1]).

---

[1] For example, it matters that the target information gets to the user, but it does not matter whether that is via wireless USB, RS232, wired USB, or some other method.

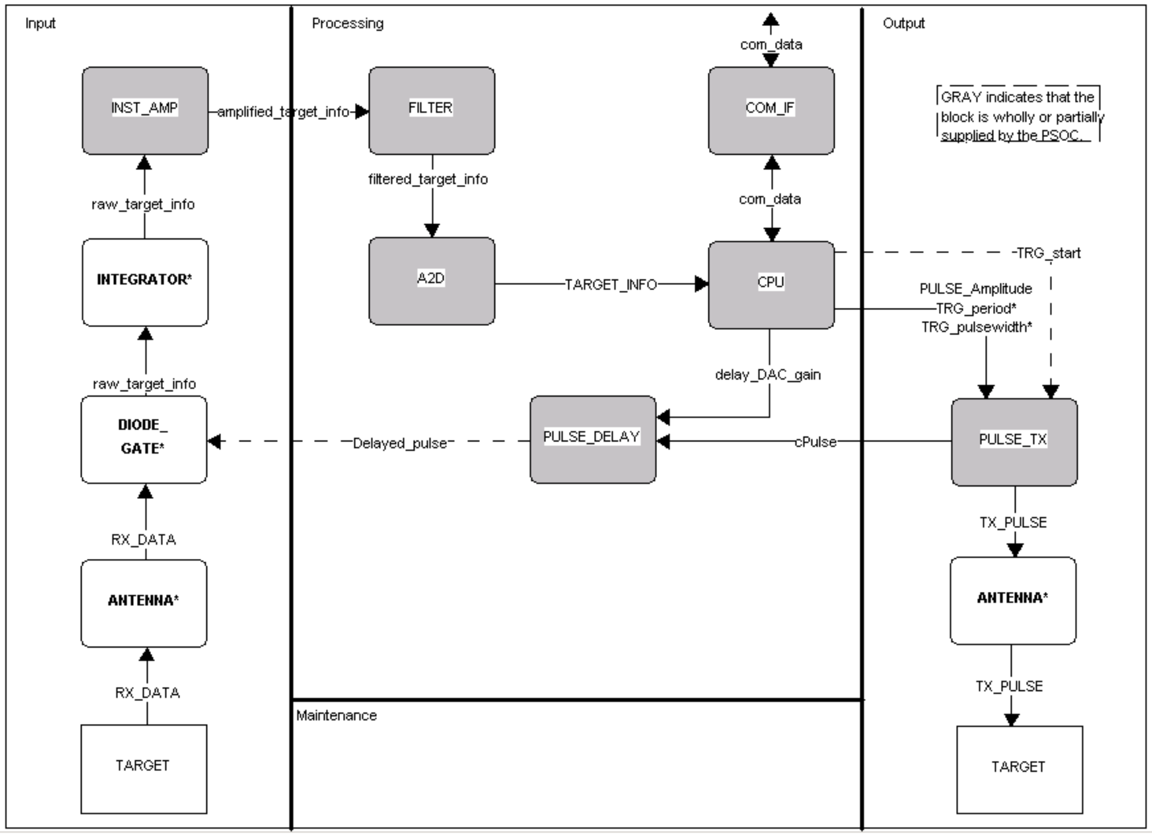


Figure 3. Top Level Architectural Diagram

Now that we can make some decisions about how we are going to do things (see Figure 3), I was hoping to be able to use the computer as a buffer between the user and the system. Let the computer save the data so that it can be processed later! Since one of the stated goals is to learn, having the data in a format that will allow the user to try different post-processing techniques is important.

The user will change the parameters using one of the computer interfaces (com_data) and then receive the target data back over the same interface.

In the pulse transmitter circuit (PULSE_TX), I wanted to have the ability to vary the transmit amplitude. In order to do this I needed a digital-to-analog converter.

In addition to varying the amplitude, I needed a way to trigger a variable number of pulses for experimentation. This part of the system needed to be very controllable. To do this I decided to use a PWM (Pulse Width Modulator) User Module. A PWM gave me the ability to control the pulse width and duty cycle of the triggering system pulses.

For the pulse delay (PULSE_DELAY), the circuit needed a way to control a variable capacitance diode used in an RC delay line formed using hardware buffers. This needed to be the best resolution, so a high resolution DAC was used to give the circuit the greatest capacitance resolution.

The receiving section uses a sampler triggered by the delayed trigger for the incoming pulses. It can be a very weak signal so it needs to be amplified (INST_AMP) to give some chance of seeing it. In addition, the incoming signal would probably need to be filtered (FILTER) before being converted to a digital signal (A2D) and transmitted back to the user.

Implementation

The PSoC made implementing the bulk of this design simple because it could be routed internal to the device.

The user sends the parameters to the system from the computer using the RS232 interface and then receives the target data back over the same interface. The RS232 serial interface was chosen in this case for its simplicity. Since the PSoC is able to implement a UART using two of its digital blocks, this was a very easy choice. This may be one of the limiting factors for receiving data, as the maximum data speed attainable on most PCs is 115.2 Kbps (14,400 bytes per second). I could have chosen to use some booster software (shsmod) to attain higher speeds on the PC side, and a faster UART-to-RS232 conversion chip, but being on a low budget I used what I had available. The current MAX232 chip that does the UART-to-RS232 signal conversion limits the data rate to not much more than 115.2 Kbps. In future designs, perhaps a USB interface would be appropriate.

The RS232 speed actually limits the sample rate on the ADCs to about 5,760 samples per second for 2-byte samples and 2,880 samples per second for 4-byte samples. Throw in a few bytes for formatting (just a CRLF) and we are now limited to 2,880 or 1,920 samples per second. That is good enough for the slow moving signals of interest.

Varying the amplitude of the transmit pulse within the supply range was simple. In order to do this, a way to convert my digital requests to a voltage in the analog world was needed. The PSoC provided a 9-bit DAC with an inverting amplifier to get the fullest range and fulfilled this need. Eventually the system will need a larger voltage (~12-20 volts) on the transmit side, so the DAC will control this as well (with a little circuit to buffer it).

To be able to trigger a variable number of pulses for experimentation, I used two of the CY8C27443’s digital blocks to form a 16-bit PWM. I had to have a way to limit the number of pulses, so I first modified the interrupt routine to count them and then stopped the pulse train when the requested number of pulses was completed. The minimum number of pulses was limited by the interrupt latency, so I looked into a way to do this totally in digital blocks. I used a counter driven by the PWM output and with the terminal output routed to the PWM enable.

In addition to this, the 16-bit PWM combined with a fast 48 MHz CPU clock gives the system the ability to vary the pulse width and duty cycle with great resolution. Pretty neat! This increases the pulse rate over the circuit I started with and makes it possible to detect faster moving targets.

These pulses trigger the transmitter to send a pulse. There are a lot of ways to generate a pulse. The shortest pulses so far are on the order of 100 Atto-seconds[2]. One method involves the use of a mismatched transmission line. With one end terminated, when a rising edge is sent into the transmission line, the reflection cancels the input signal. For example, a 100-foot long cable gives a delay of about 300 nS. Another method is to use a step recovery diode[11]. The step recovery diode can give 0.8V pulses in the 100 pS range at a reasonable cost. Another possible method is to use a Limited Space Charge Accumulation Diode. This device (similar to a Gunn diode) can generate pulses on the order of 7 nS with overtones carrying some usable power out to about 1 pS.

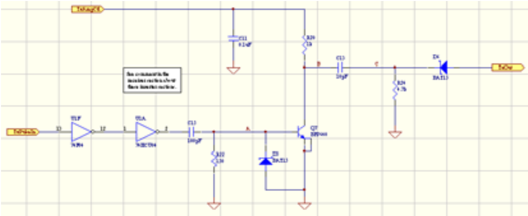


Figure 4. Pulse Transmitter

The circuit in this project (Figure 4) uses an RC differentiator in front of a high frequency transistor (FT~70 GHz). In its current incarnation I did not need a transistor that fast, but wanted to get experience with it for future work in this area. There is a small capacitor on the collector of the transistor. When it is not conducting, this capacitor charges up to the supply at a rate limited by a 1 kΩ and 4.7 kΩ resistor. The pulse is generated when the transistor goes into conduction and shorts the capacitor to ground. It discharges quickly and generates a pulse. The extra inverter sections in the transmit section serve two purposes. They enable the transmitter to accept the positive going pulses, buffer the transmit section, and equal out the delay between the transmitter and receiver to allow detection at shorter distances.

---

[2] Yep, I had to look it up, too. 1 Atto-second = 10^{-15} seconds.

The pulse delay circuit (PULSE_DELAY) uses a 9-bit DAC from the PSoC analog blocks to control the capacitance of a variable capacitance diode. Because this needed to be the best resolution, a 9-bit DAC was used to give the circuit the greatest capacitance range.

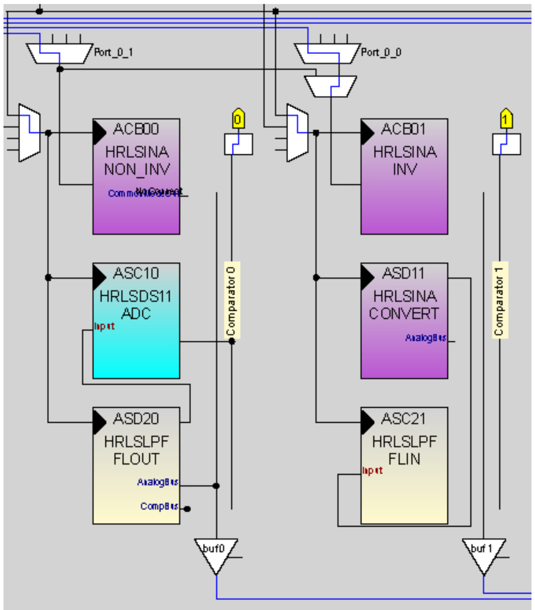


Figure 5. Receiver Amplifier/Filter Line

The transmitted signal travels from the antenna and an attenuated echo returns to the antenna system. If the target is not at the expected distance, the system ignores it. If the target is at the correct distance, the return signal is detected by the diode gate and averaged by the integrator section before being handed over to the PSoC.

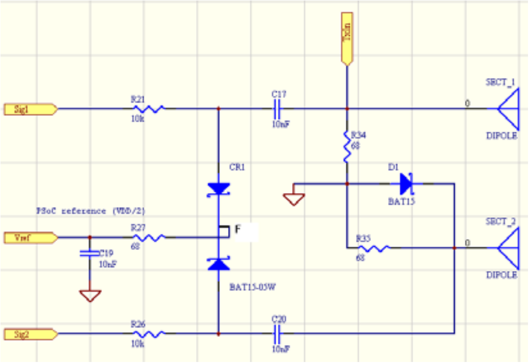


Figure 6. Diode Gate Sampler/Integrator

The receiver section in Figure 6 deserves more explanation because it is where the magic occurs.

As a signal is impressed onto the antennas, it appears on the antennas and thus the 68Ω resistors. When a valid echo appears and the sampling diodes are pulled low by the delayed pulse at point ‘F’, the diodes conduct and the signal is dumped into the input capacitors. When the pulse releases the cathodes of the gate diodes, the signal appears at the signal inputs. This increases the signal-to-noise ratio by recording the voltage only when the signal is present, and ignoring time periods when there is no signal and only noise.

The returned echoes are very weak and in the presence of background noise, so one pulse is not enough to get the proper signal. The 68Ω/10 nF combination is much faster than the 10 kΩ/10 nF combination and therefore the capacitor contains a charge proportional to the average of the returned echoes. This gives a gain to the signal in the presence of noise. To illustrate, imagine yourself in an early eighties disco bar. The strobe light is pulsing madly. You catch a glimpse of the person you know will be your soul mate in one of the flashes. If that person stands still, you can see them clearly. If not, you may lose that person in the “noise.” You don’t have the option of turning on the main lights, but by increasing the rate of the flashes you can see this person better even if they are moving. Your eyes are in effect averaging the signal with a high pulse rate. This is process gain.

In this project, process gain is modeled using a Poisson distribution. I have not yet implemented jitter, but it can still be thought of that way. As explained in [2], the time constant for a significant change in the RC filter network is about 2π(10 kΩ)(10 nF) = 628 µS. The system can have a pulse rate from the PSoC’s PWM as fast as 12 MHz, so the average number of received echoes in this time period is (628 µS)(12 MHz) = 7,536. The signal to process gain, or signal to noise improvement, is given by 10Log(7436) = 38 dB of gain.

I’m not an antenna expert, but I think the antennas receive the signal 180° out of phase. If this is true, because the instrumentation amplifier measures the signal difference, we get another 3 dB of gain. The instrumentation amplifier is a stable, high gain amplifier and has differential inputs. It is designed to ignore common mode noise gaining an advantage over a single-ended system.

The receiving circuit design called for the input signals to be amplified (INST_AMP), filtered (FILTER), and converted to digital (A2D) for transmission to the user. Accomplishing this on the PSoC (Figure 5) took some experimentation because I wanted to be able to route the output of the amplifier and filter to the outside so that the signal could be looked at on a scope. I chose the 3-block instrumentation amplifier because I am an engineer and more is better right? Actually, the improved common mode rejection and greater amplification looked attractive and turned out to be necessary.

Using the Filter Design Wizard, I spec-ed out a low pass filter using another 2 analog blocks. Here is where the flexibility of the device really helped because when the signal could not be routed the way I needed, a different low pass filter topology fixed the problem.

Since I wanted to be able to detect aspects of the returned pulses from different materials, I needed not only decent conversion speed, but I needed to have a mode with good resolution. I also needed to be able to go from one end of the resolution/speed spectrum to the other. The PSoC dynamic re-configurability helped with this because I could choose a fast A2D with lower resolution or a slow A2D with greater resolution. I chose both!

The PSoC device and PSoC Designer 4.0 allow the use of Dynamic Re-configuration. This makes it easy for the user to include several different, and possibly related, functions in the same device. In this device, I implemented a configuration with a high resolution, but lower speed **and** a configuration with a higher conversion speed, but lower resolution.

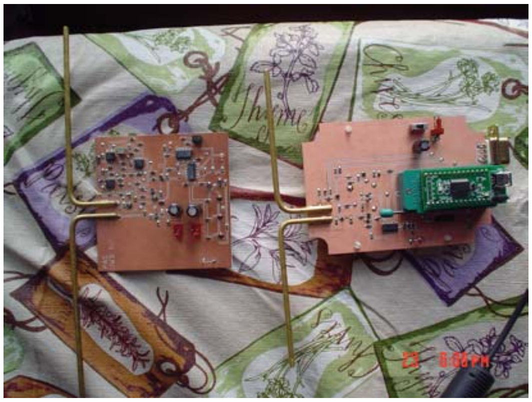


Figure 7. Old and New Implementations[3]

Figure 7 shows an implementation of the original circuits of the type described in UWB RADAR patents. It works, but had low functionality, was difficult to tune in, and could only detect moving items. The circuit on the right has greater functionality, is easier to tune in, and can be made to detect moving items (my next addition). It is also faster, smaller, and uses much less parts (about 45 vs. about 75). Imagine it without that big, old ZIF socket in there!

Some obvious things still need to be implemented:

- Add noise modulation to the pulse train[4].
- Transmit amplifier DAC needs to be sent to a circuit that will make the signal go between say 3.5 to 20 volts for more output power (design level SNAFU).
- UART components need to be replaced with something faster or the COM channel upgraded to USB.
- Add programmable gain for the signal pathway to the command structure.
- Add programmable sequences for the pulses.
- Add ability to slew the delay to detect non-moving items.
- Escape a control character.
- Add a local switching power supply.
- Add a Windows client to handle the control and data.

There are a lot more ideas I have, and the list is rather large and I won't bore you with it. Let it suffice to say that I could have filled two or three columns with these improvements and ideas.

I would have to say that overall, the PSoC device shortened the design cycle considerably. It would have normally required a lot more parts and I was able to concentrate on the application level.

---

[3] Yeah, yeah, I did this at home on the kitchen table!

[4] I decided to save this one for later to give me something to do this weekend.

Performance

I love this part! Debugging and checking out the performance of any system can be difficult, but it is almost always rewarding. I had access to a small oscilloscope for basic debug and was able to take a few snapshots of the faster pulses on a better LeCroy scope to check out the performance of the transmitter and delay/receiver line(s).

The first thing any good project should do is blink an LED. This gives you something visual to see without using a scope, and lets you bridge the gap between getting something to work and *seeing* something work. Doing this was a snap.

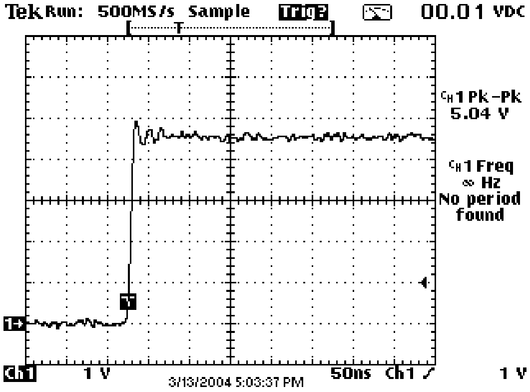


Figure 8. LED with Long Period (STRONG)

This also gave me the chance to see the speed of the rise times on the PSoC (Figure 8). This could affect the ability of the receiver and transmitter, but I wanted to verify the values from the data sheet. The scope with which I took this reading had insufficient horizontal (time) resolution to really get a good look at it, but I tried my LED “visual watchdog” out with both the STRONG and SLOW STRONG settings.

The CY8C27443 documentation lists rise times of 3-18 nS with a 50 pF load. This does not have anywhere near a 50 pF load. In addition, the horizontal resolution on the scope used is not good enough to show the fast rise times (Figure 8). Anyway, it looks about right (best guess is about 5-10 nS – okay, considering the ZIF socket capacitance).

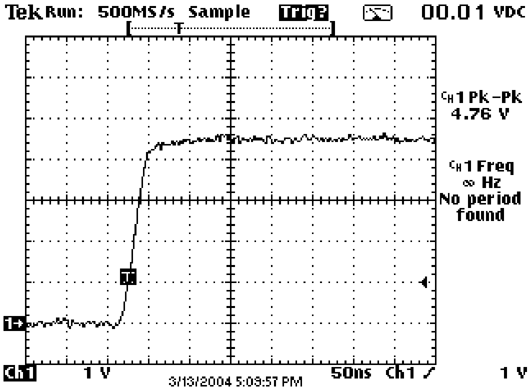


Figure 9. LED with Long Period (SLOW STRONG)

This is interesting and worth notice. If you want to reduce the emitted noise, just compare the two slew rates (Figure 9 vs. Figure 8). Notice how much less noise/overshoot there is with the SLOW STRONG setting! I left the LED on the SLOW STRONG setting while the pulse trigger line got the STRONG setting.

The software for the communications was not difficult since the PSoC UART libraries already have a method to collect commands. I used this and soon had the UART up and running.

Getting the pulse train working took some effort. As stated earlier, at first I just decremented a variable in the interrupt of the PWM. This was fine for slow, large numbers of pulses, but when the pulses were few and short, the interrupt latency got in the way. This is a common problem with all microcontrollers.

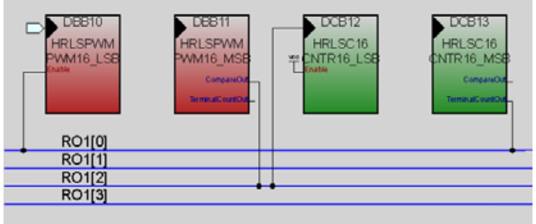


Figure 10. PWM Pulse Counter

PSoC digital blocks to the rescue! Click, click, click and I was done. I just added a counter using the PWM as its clock and ran its terminal count into the enable on the PWM. Presto, an instant pulse counter with automatic disable (Figure 10).

The signal chain output at the low pass filter was the next goal. Could I actually send pulses and get back the return signal on the output of the low pass filter?

I wrote a little more code and added the ability to trigger a continuous train of pulses (versus a set number of them) out the transmit chain. There were a few mistakes in component values[5], but after a little effort the transmit pulse looked great.

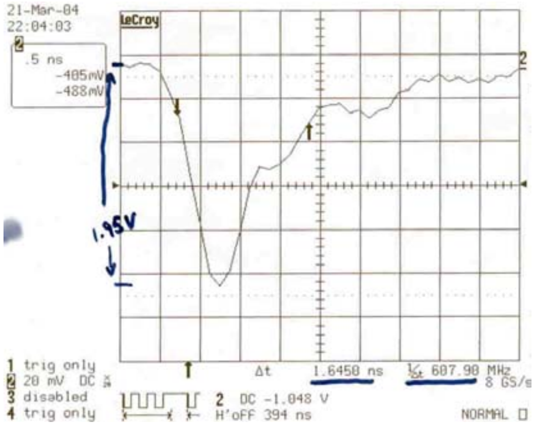


Figure 11. Transmit Pulse

As you can see here, the transmit pulse is about 1.95 volts deep and 1.65 nS wide where I measured it.

The transmit pulse was really starting to look good so I started on the receive side. I had taken some readings of the receive delay versus the DAC setting so I knew that this was working. (See Figure 16. Delay versus DAC Output in the Appendix.) The pulse to the sampler in the receiver was a little weak so I boosted the coupling cap from 0.5 pF to 5.6 pF and this fixed the immediate problem. With a little more effort (less than the transmit side), the sampler pulse was soon looking good and in the range needed.

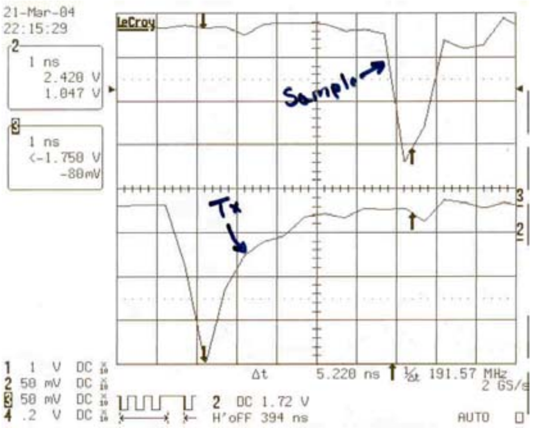


Figure 12. Receive Sample and Transmit Pulse

As you can see, the transmit pulse is about 5 nS behind the delayed receive/sampler pulse. Right on the money!

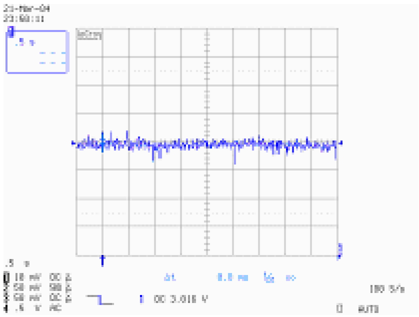


Figure 13. Output of LPF with no Motion

It was now time for the big test! First, set the pulse width parameters and voltage output on the transmit amplifier. Then start the pulses. I verified everything was quiet on the signal pathway (Figure 13) and attached a scope to the low-pass filter output. I then wiggled a section of PCB in front of the antenna at about the correct range. It turned out to be a little less than predicted, but this was probably due to scope probe capacitance when I measured the transmit and receive sampler signals. The capacitors in these areas are on the same order of magnitude as the scope probe.

---

[5] This is also where I realized that more than half the DAC output on the transmit amplitude is wasted. Should have seen this before! Lucky this was a 9-bit DAC.

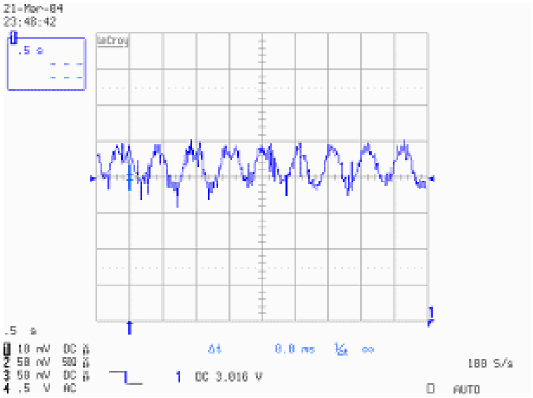


Figure 14. Output of LPF with Motion

I saw a very small, but definite wiggle (Figure 14)! I was on my way and what a relief to get a signal.

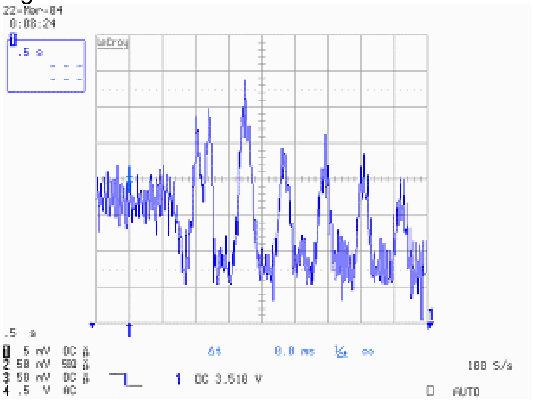


Figure 15. Output of LPF with 5 dB of Gain

Click, click and I added a little gain to the receive signal chain and got a stronger return signal. In Figure 15, you can see where the line was “quiet” and then where I wiggled the PCB section.

There is still a lot to check out, but the system is basically functional. With a little bit of TLC it will soon be ready for use in evaluating the various parameters of a UWB system. I even sat in front of the unit in my rocking chair and watched the signal go back and forth.

Appendix

Appendix A - Bill of Material

Q	Part	Designator
===	============	==========
1	CY8C27443	U6
1	74HCU04	U1
1	LM3405	U4
1	MAX232CWE	U7
1	100uF	C26
1	10uF	C4
5	1uF	C27 C28 C29 C30 C31
4	0.1uF	C11 C12 C21 C5
4	0.01uF	C25 C17 C19 C20
1	100pF	C15
1	15pF	C18
1	10pF	C13
1	5.6pF	C16
3	10k	R21 R25 R26
1	4.7k	R24
2	1k	R20 R8
1	750	R19
1	390	R101
2	150	R22 R23
3	68	R27 R34 R35
1	BAT15-05W	CR1
4	BAT15	D1 D3 D5 D6
1	BBY56-02W	D2
1	LED	D101
2	BFP640	Q7 Q8
2	1/8” BRASS	SECT_1 SECT_2
1	SPDT	S1
1	CONNECTOR	J3
1	CONNECTOR	J1
1	DB9	J5

Appendix B – Throughput Bottleneck Calculations

Limit	115,200	bps

BYTES	PSoC ADC SAMPLE RATES (Max Sa/s)									
	2	**4**	**7.8**	**120**	**250**	**480**	**7800**	**10000**	**32000**	**40000**
1	20	40	78	1200	2500	4800	78000	100000	0	0
2	40	80	156	2400	5000	9600	0	0	0	0
3	60	120	234	3600	7500	14400	0	0	0	0
4	80	160	312	4800	10000	19200	0	0	0	0
5	100	200	390	6000	12500	24000	0	0	0	0
6	120	240	468	7200	15000	28800	0	0	0	0
7	140	280	546	8400	17500	33600	0	0	0	0
8	160	320	624	9600	20000	38400	0	0	0	0
9	180	360	702	10800	22500	43200	0	0	0	0
10	200	400	780	12000	25000	48000	0	0	0	0

bits per Byte	10	bpB	Clock rate multiplier		8
Bytes per sample	3	BpS	Sample clock input		30,720
bits per Sample	30	bpS			
bits per second	115,200	bps	CPU	24,000,000	48,000,000
Samples per second	3,840	Sps	Divisor	781.3	1,562.5

Rate	115200
bits/Byte	10

RS-232 MAX SUPPORTED SAMPLE RATE			
BYTES	1	2	3
MAX SPS	11,520	5,760	3,840

Appendix C – Receive Delay Line Values (Done Very Early Before Some Circuit Changes)

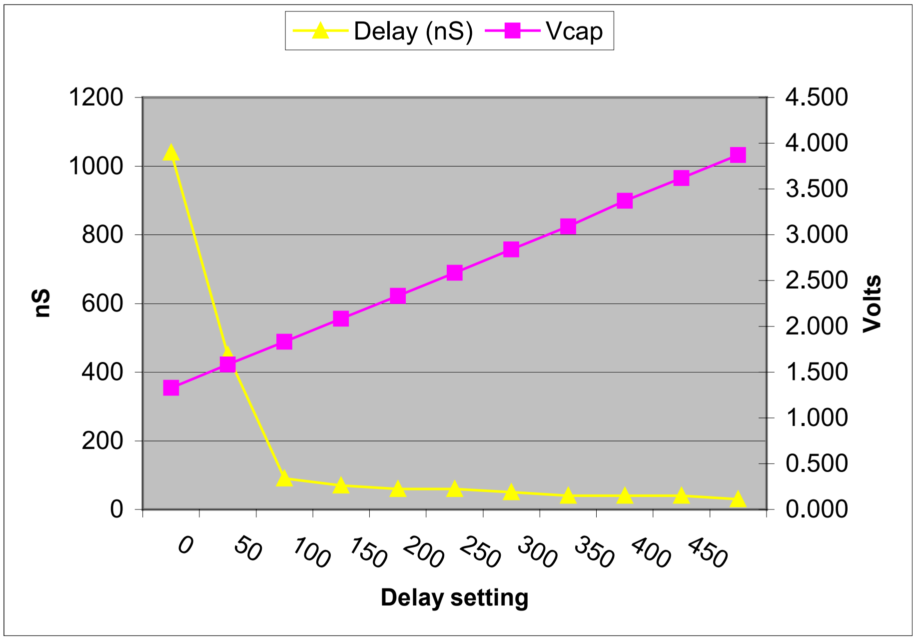


Figure 16. Delay versus DAC Output

Appendix D – Low Resolution, High Speed Setup

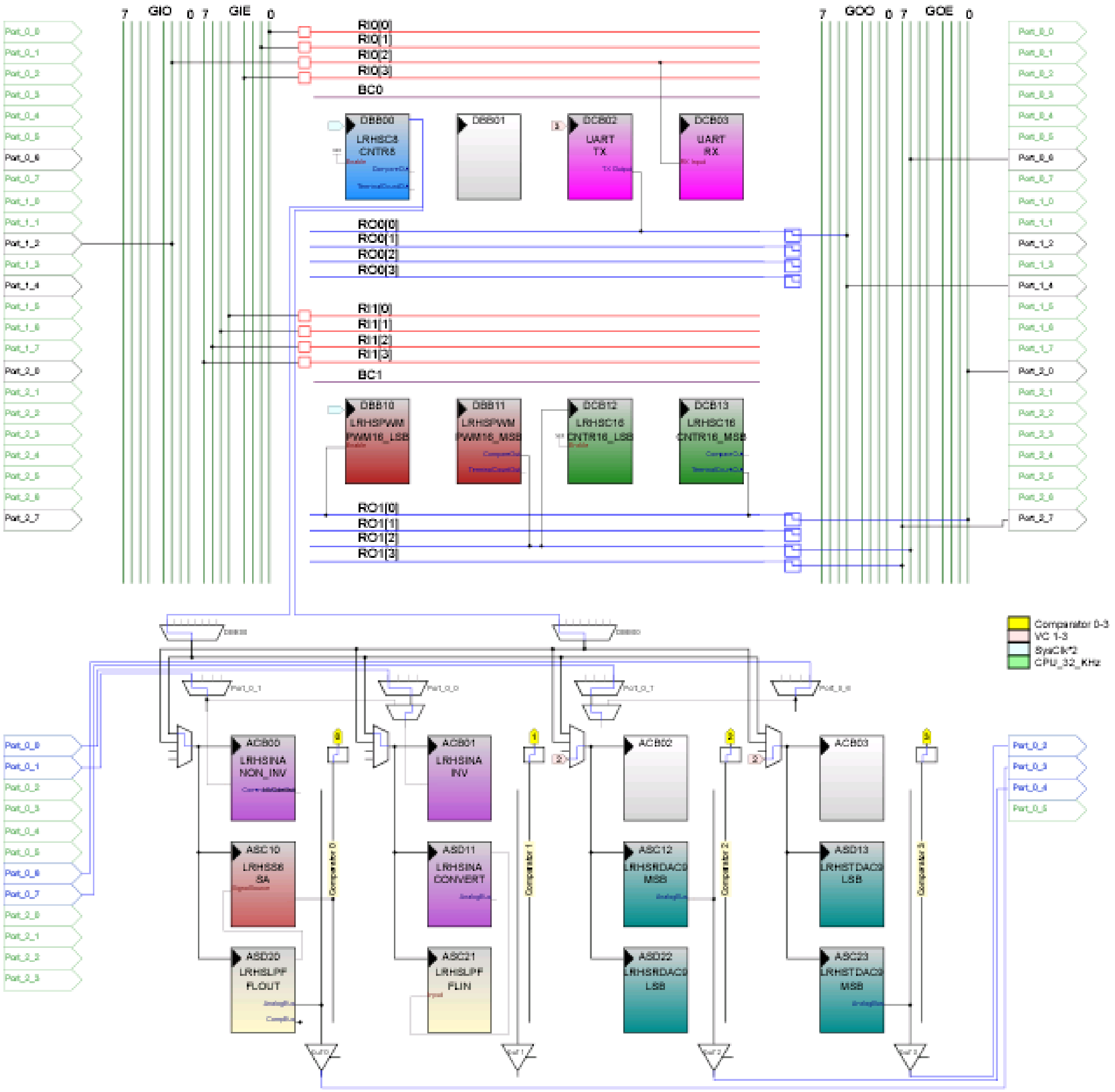


Figure 17. Main Configuration (Low Resolution - High Speed)

Appendix E – High Resolution, Low Speed Setup

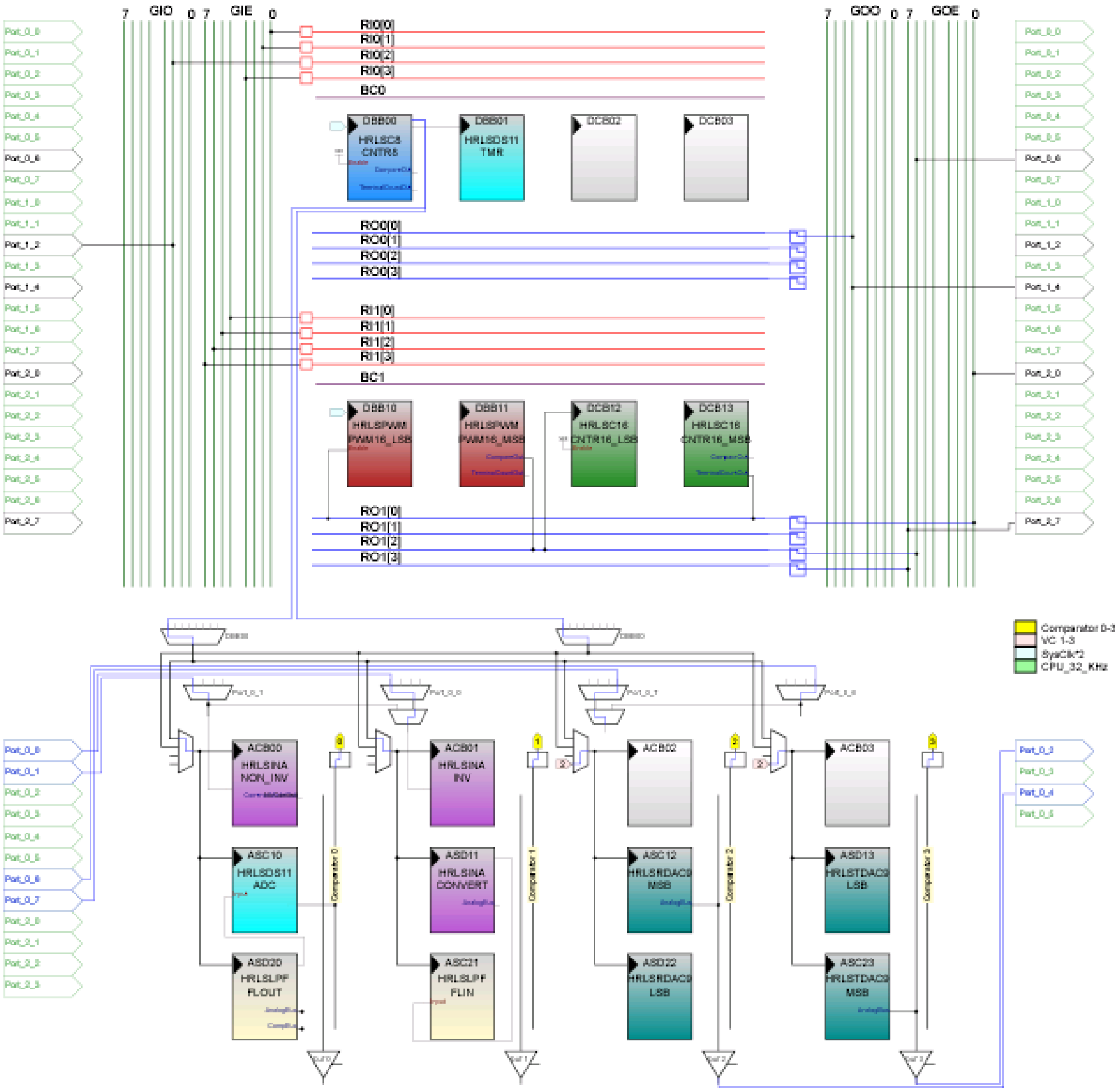
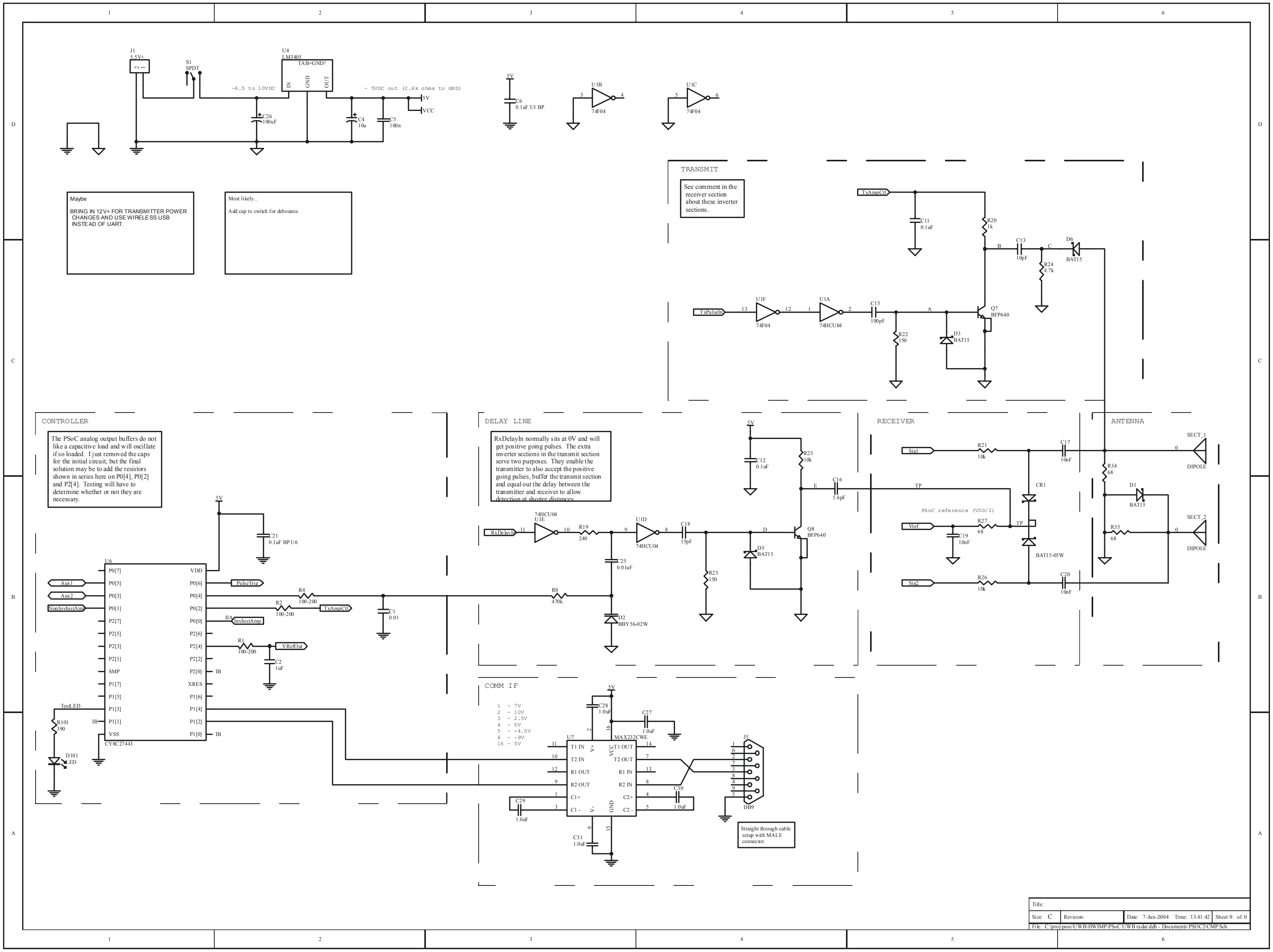


Figure 18. High Resolution, Lower Speed Configuration



The Radar Schematic

Acknowledgements

Patents with Sources

[1] http://www.uspto.gov/, McEwan T.E., (1994) *Ultra-wideband radar motion sensor,* United States Patent 5,361,070.

[2] http://www.patent.gov.uk/, Watson, R. A. W., (1937) *Radiolocation,* Great Britain patent GB593017

[3] http://depatisnet.dpma.de/, Huelsmeyer's C., (4-30-1904) *Verfahren, um entfernte metallische Gegenstände mittelsche elektrischer Wellen einem Beobachter zu Melden*, German patent DE0165546

[4] http://depatisnet.dpma.de/, Huelsmeyer's C., (11-11-1904) *Verfahren zur Bestimmung der Entfernung von metallischen Gegenständen*, German patent DE0169154

[5] http://www.uspto.gov/, Huelsmeyer, C., (1-16-1906) *Telemobilescope*, US patent 810,150

References

[6] http://www.techonline.com/, Staderini, Enrico M., *Everything you always wanted to know about UWB radar… : a practical introduction to the ultra wideband technology*, Online Symposium for Electronics Engineering, 2000

[7] http://www.4reference.net/encyclopedias/wikipedia/Radar.html, RADAR notes and history

[8] http://www.radarworld.org/huelsmeyer.html, RADAR notes and history

[9] http://www.penleyradararchives.org.uk/history/index.htm, History of Watson-Watt's British RADAR

[10] http://nrotc.wisc.edu/battalion/courses/Course%20Index.htm, ROTC RADAR course

[11] http://www.ntu.edu.sg/home/EYWYeap/, Generating a Pulse With a Step Recovery Diode

Victor Kremin, Aaron Mathew, William Moretto, Mohana Koteeswaran, and Paul Colohan from Cypress Applications Support for their support during this project. They answered all eight knowledge base cases to my satisfaction.

About the Author

Name: Peter A. Stephens
Title: Senior Research Engineer III at Delphi Systems
Background: BSEE from the University of Michigan, MSRE from the University of Maryland.
13 years in engineering at Delphi in Kokomo, Indiana.
Contact: peter.a.stephens@delphi.com

Cypress MicroSystems, Inc.
2700 162nd Street SW, Building D
Lynnwood, WA 98037
Phone: 800.669.0557
Fax: 425.787.4641
http://www.cypress.com/ / http://www.cypress.com/support/mysupport.cfm