5-2018

Software Defined Radar For Vital Sign Detection

Chandler J. Bauder
University of Tennessee, Knoxville, cbauder@vols.utk.edu

James Bates
University of Tennessee, Knoxville, jbates13@vols.utk.edu

Steven Engel
University of Tennessee, Knoxville, sengel1@vols.utk.edu

James S. Tucker
University of Tennessee, Knoxville, fgg211@vols.utk.edu

Fangzhou Liu
University of Tennessee, Knoxville, fliu19@vols.utk.edu

Recommended Citation

Bauder, Chandler J.; Bates, James; Engel, Steven; Tucker, James S.; and Liu, Fangzhou, "Software Defined Radar For Vital Sign Detection" (2018). *University of Tennessee Honors Thesis Projects.*
https://trace.tennessee.edu/utk_chanhonoproj/2219

Software Defined Radar For Vital Sign Detection

Team #23: James Bates, Chandler Bauder, Steven Engel, James Tucker, Fangzhou Liu
Customer: Dr. Aly Fathy

April 24, 2018

Table of Contents:

Executive Summary:

This product is an expansion of one of Dr. Fathy's current research projects. The current implementation uses a transmit and receive chain that is made up of bulky, discrete components that sends and receives a signal to measure the vital signs of a patient with no contact. These components prove to be costly, totaling around $3000. The need addressed for our project is to implement this radar wish Software Defined Radio (SDR) in order to reduce the size and cost of operation.

In general, this product has numerous useful applications. The idea for the research project originally came from interest from the army. In a search and rescue mission, it would be beneficial to know if the captive is alive to determine if a team should be sent in. Other applications involve situations where contact vital sign detection techniques are not ideal. If a patient is badly burned, it is difficult to apply contact-based vital sign detection sensors without causing further damage or discomfort. It could also be used for the elderly who require constant monitoring, but may often forget to wear monitoring bracelets or prefer to not wear intrusive devices. In disaster situations, it could be used to take the vital signs of a person trapped in rubble.

Our goal is to create a system capable of detecting and monitoring various patient vital signs without making contact. Our system should be able to monitor heart and respiratory rates from approximately 1 meter without making any contact using Continuous Wave (CW) Radar methods, meaning using only one frequency. Using previous designs for similar systems, we will focus on improving the design to reduce cost and increase speed, while aiming to improve accuracy over earlier designs. While previous designs used discrete components, there are several limitations and extra costs to this approach, something we hope to improve upon through the use of software defined radar. Instead of having hardware to generate, transmit, receive, and process a specific signal or set of signals, our system will have hardware that can be fine-tuned by software to transmit and receive any number of combinations of signals within a certain range. Furthermore, instead of having dedicated processing hardware, which is often costly, we will do the signal processing through software on general purpose hardware. In addition to allowing for quick changes in how the data is processed, this approach will drastically reduce costs and size. We also originally planned to implement Stepped Frequency Continuous Wave (SFCW) Radar methods in order to have functionalities such as tracking.

Our team was successfully able to reach the functionality of the hardware system with discrete components through software defined radio. We managed to successfully read vital signs of people in real time and display the results. Measuring respiratory rates from 1 meter away proved to be especially accurate, but no more accurate than the original system. Heart rates were harder to read. By using an amplifier and having the patient hold his or her breath, the heart rate was noticeably present. However, this is not a reasonable method for vital sign detection in the real world. We were also not able to implement SFCW as we had hoped. Limitations of the SDR system we are using prevented us from getting accurate results.

Most of our challenges came with discovering the functional abilities of the SDR system and switching the signal processing from MATLAB to the Raspberry Pi. The SDR limitations prevented us from meeting our goal of implementing SFCW, though we were able to meet our preliminary goal of using CW to measure the vital signs. Learning how to convert the signal processing code of MATLAB into python that the Raspberry Pi could use was also another challenge we faced. We had to find a library with mathematical functions similar to MATLAB in order to get the same results using both signal processing methods.

Requirements:

1. SDR Hardware Selection
   - 1.1. Hardware must be able to function properly from 2 to 4 GHz
   - 1.2. Must be able to quickly transition from transmitting one frequency to the next
     - 1.2.1. 100 steps of 20 MHz (2 GHz) at a rate 50 ms or less for a full sweep
   - 1.3. Must be controlled by software code for easy changes
     - 1.3.1. Hardware must be capable of running lightweight Linux distributions is ideal
   - 1.4. Hardware should have a simple and friendly User Interface
     - 1.4.1. Hardware should be capable of displaying, directly or indirectly, the results of the vital sign measurements.
     - 1.4.2. Must provide a visual output either through a connected display or by connecting an external display through HDMI.
     - 1.4.3. Hardware should be capable of receiving user input, either through on-board buttons (start/stop measurement and similar functions) or through external connection options for I/O devices.
2. Software Selection
   - 2.1. Must allow for the fastest communication between software and hardware
     - 2.1.1. Low software, language, and API overhead is required to prevent our system from taking longer than the original system during data acquisition
   - 2.2. Must be highly customizable
     - 2.2.1. Must be able to handle a change in application for the software-defined radio to support future endeavors in tracking and medical imaging
     - 2.2.2. Must be able to change and update data acquisition techniques from one-time runs to real-time acquisition and display
   - 2.3. Must be capable of real-time vital sign detection for delay-free data acquisition and processing
3. Abilities
   - 3.1. Measure the respiratory rate of a subject who is approximately 1 meter away

3.1.1. Must be able to measure a respiratory rate of 12-20 breaths per minute (0.2-0.33HZ) with an accuracy of ± 5%

3.2. Measure the heart rate of a subject who is approximately 1 meter away

3.2.1. Must be able to measure a heart rate of 60-90 beats per minute (1-1.5 Hz) with an accuracy of ± 5%

3.3. Display corresponding respiratory rate and heart rate results on the controlling computer or display

Requirements in Depth

1.1 2-4 GHz is a range of frequencies that is used to accurately detect objects in high clutter environments, such as the human body. Because the human body contains materials of many different densities, a wide frequency range is necessary to obtain all the data at the required resolution that we need. Thus, our SDR must be able to use this range of frequencies.

1.2 If the transition from frequency to frequency is too slow, too much information is lost. This becomes a factor during object movement. If the scanner takes too long between each step, we cannot accurately fit all of the data together.

1.3 One of the major focuses of this project is flexibility at lower cost. Systems based on discrete components have a rather fixed behavior, meaning any changes would be costly and time intensive.

1.4 As this system would ideally be a medical imaging device and possibly a field use tool for search and rescue, making the output of the system readily visible and easy to interpret is necessary. Use of the finished system should not require a degree in Electrical Engineering or even knowledge of signal processing. The user interface should be kept small and lightweight to keep down on overall size and cost.

2.1 Reducing overhead from sloppy code and inefficient languages is important to keep the system operating in a Firm real time type mode. If the system takes too long on any one task, data can start to be lost, reducing the accuracy of the system.

3.1 / 3.2 Accuracy of ± 5% was deemed necessary but adequate based on both the application of the system and the accuracy of other current methods for measuring heart rate.

Changelog:

11/9/17 - Amendment to 1.5.1: System should be able to display readings in real time using either on board displays or an external display connected through HDMI or other display protocol.

11/9/17 - Addendum to 1.1 and 1.2: Similar but cheaper hardware with a somewhat reduced feature set should also be explored, to allow for cheaper/less complex models to be designed for a wider potential consumer audience.

2/5/18 - Addendum to 1.4: Use of an external system to capture, process, and display the data should be explored. A Raspberry Pi, or other similar device, should be able to capture the data via Serial connection, process, and display it quickly. Such a system would reduce development time that would otherwise be spent writing overly complicated code for the Zedboard platform.

3/16/18 - Addendum to 1.1: With regards to the antennas, preference should be given to cheaper and smaller antennas that can operate in this frequency range and still meet the needs of the system. The lab provided horn antennas are bulky and expensive, diminishing the cost-saving aspects of this project.

Design Process:

- Describe how the effort was decomposed into manageable pieces to address the agreed upon requirements.
    - Our project relied heavily on having multidisciplinary team members with a background of electrical engineering as well as computer science and computer engineering. It required knowledge from both the fields of electrical engineering and computer science, so it made sense to split the project into two areas to focus on: hardware and software. Both sides worked together, but the major decision making on each side was made by those with the specialized backgrounds who knew what was best for the team. This made the work flow very smooth and no one was lost or was left working on a part of the project for which they were not very qualified.
    - To also assist in managing the pieces of our project, we only tackled one obstacle at a time. Rather than looking at all of the problems at once, we decided to break problems down into specific areas that they address and focus on them at as low a level as possible. This really helped pinpoint our focus to get the most efficiency out of our time when we met up each week. The weekly meetings also helped us manage all the different pieces of our project and coordinate the work being done by different members to make sure that everyone was on the same page at any

given time. The same can be said for the team lead's weekly meeting with the customer.

- Describe any standards your group may have adhered to while working on your project.
    - Because we are sending and receiving data wirelessly, we had to make certain that we were compliant with FCC regulations on wireless data transmission. The FCC assigns certain frequency bands for use by different groups of people, and we must keep these restrictions and regulations in mind. Another standard we had to consider is that the FCC also has regulations on how much power you can be transmitting within a certain range of civilians. Our project focuses on patient safety, so we would never want to transmit a greater amounts of power than is safe for human exposure, but we still want our signal to be as powerful as can be so that we can obtain clearer results.
- What open questions had to be answered by research?
    - In order to decide what SDR platform to choose for our project, we had to research what the desired frequency range is for vital sign detection. We learned that we are much less limited when it comes to frequency when dealing with vital sign detection as opposed to medical imaging. With the current system, a 915 MHz signal was adequate enough to detect a strong vital sign signal. This is relatively lower than the desired frequency range for imaging and tracking, which have a desired frequency range of about 2 to 4 GHz. In fact, for vital sign detection, there is only a certain frequency that we cannot go over in order to get accurate results. This frequency is well out of our considered range, so we do not need to take it into consideration. We also had to research capabilities of different components and previous research done in the field in order to gain a grasp of what has been done and what we might want to try and do.
    - To select the software portion of our project, our main consideration was the ability for the software option to sweep through frequencies at a certain speed. This required researching what the slowest performance allowed is for the desired software. If this project is to expand our project to imaging and tracking for further research after we are done with it, it would need to sweep from 2 to 4 GHz through 100 steps in less than 50ms. This means software and hardware must be able to switch from one transmitting frequency to the next in no more than 2ms.
- What alternatives were explored?
    - An alternative hardware solutionis the PlutoSDR. We knew we wanted to stay with Analog Devices products, and this is a brand new option on the market for us to consider. It is built to target students learning about software defined radio, so it is more user friendly than the AD-FMCOMMS4-EBZ. It also helps that this is a product that is optimized for its entire advertised frequency range. The AD-FMCOMMS4-EBZ is just an evaluation board, and made mainly for

waveform testing, though it is still more than acceptable for our application. The Pluto is also cheaper and smaller than the AD-FMCOMMS4-EBZ. However, the PlutoSDR's frequency range is smaller and would only be useful for vital sign detection, which would prevent further extension that Dr. Fathy and his research assistants are hoping to be able to explore after we are done with our senior design project. We would like to plan for the future, and the PlutoSDR simply cannot give us that flexibility. The AD-FMCOMMS4-EBZ is capable of a higher frequency range and can implement stepped frequencies for imaging and tracking.

    - Another alternative hardware solution we considered was performing all of the processing and calculations on the ZedBoard, but we decided to expand to a Raspberry Pi to handle the processing and also the display. We made this decision because the ZedBoard was limited in its ability to multithread and it would have been a difficult, if not altogether impossible, task to get it to read in data and process it simultaneously or within a period of time small enough to be acceptable for our applications. Using the Pi also weaned our dependence from having a big bulky computer attached to our system.
    - We also looked over a few possibilities with our choice of software. We had the initial decision of writing a Matlab script on a computer or using a program on the ZedBoard called GNUradio. Both had decent graphical user interfaces, are well-documented, and can be loaded onto the analog device. After deliberation we decided instead of either of those options to write directly to the ZedBoard using C code. When we made the decision to use the Raspberry Pi for processing we decided to write to the Pi in Python.
- Describe the selected solution
    - Our requirements were fairly straightforward and thus it was easy to determine if our selected solution fit our needs. In simple terms, the system has to be able to sweep through a number of frequencies to transmit and receive on to be able to take measurements in several different applications, the system should be low cost to make it competitive with the current experimental device which uses expensive discrete components, and we should attempt to make the new system much smaller and more portable than the old one. We saw early on that the best way to address these requirements would be to split our focus to both software and hardware based components.
    - For the software aspect, we focused on being able to run the program(s) quickly and efficiently. The biggest question we needed to answer was what language was going to give us the best performance overall, and what tools and environment we should use to write our programs. From the research already performed, we saw that some software implementations are not ideal for stepped frequency due to the changing variables.

- The best option, we feel, is to implement our program on the ZedBoard using C code. The hardware we are using (see following paragraphs) is a mixture of an FPGA and an ARM processor. The other advantage to C code is that it is very fast and efficient. Other software options offered attractive features or ease of coding but lacked the performance only obtainable using C. While C may not always be the first choice for signal processing, there are a number of open-source signal processing libraries written in C, such as liquid SDR, and the hardware selection will allow for a certain amount of pre-processing we can take advantage of by writing code in C. We also used the Python language when writing to the Raspberry Pi.
- With regards to hardware, we decided our biggest consideration would be keeping cost low while still maintaining a large degree of freedom and flexibility for future improvements and features. These requirements fell beautifully in line with the Raspberry Pi. Its low cost and powerful processing power made it an excellent choice for filling in the gaps where the ZedBoard was not powerful enough to perform the necessary tasks on its own. It was important that our system have a lower cost than a system consisting of discrete components, but we had to analyze and question how far we need to go for our lower cost to be meaningful, and what features, if any, we could sacrifice to meet this goal.
- After weighing the pros and cons of a few different options, we decided that the Zedboard paired with an AD-FMCOMMS4-EBZ transceiver was the best option. The Zedboard, with a high-end Zynq chip from Xilinx at its heart, offers plenty of processing power for us to work with, while the AD-FMCOMMS4-EBZ from Analog Devices has a diverse set of features and impressive specifications, including the ability to quickly sweep frequencies in our desired range of 2 to 4 GHz, which should allow for reasonably accurate vital sign detection under a number of conditions. The Zynq chip, being a mixture of FPGA and ARM, will allow for pre-processing to be done on the FPGA portions before being handed off to the ARM core, reducing the work the ARM core has to do and offering increased speed across the board. Then most of the rest of the processing after data collection will be performed by the Raspberry Pi. The other benefit of this hardware choice is that it fairly software agnostic, meaning we aren't tied down to one coding environment and could choose from a number of programming languages. While this combination of hardware wasn't the most inexpensive option considered, it still had a cost of roughly ⅓ the discrete component based solution while offering far more flexibility and features overall.

- How did the team ensure that all requirements were met (to the extent that they were)?
  - The team was well versed on what requirements were set by our customer, so everyone was easily able to contribute to selecting the right solution to meet these

requirements. The team leader also routinely met with the customer and the graduate student helping the team out in order to go over progress made and any changes that were to be made to requirements.

    - Our project needed to obtain samples fast enough to accurately measure vital signs, process that data quickly enough to give a live readout of the results, and be relatively inexpensive. While it would be nice to use the simplified and more user-friendly coding programs that are designed for SDR applications and reduce the necessary legwork involved with hard coding the entire operation of the board, but they simply do not perform efficiently enough for the uses that we require for this project, and they restrict us to very limited pathways through which we would be able to implement our project. Coding in C is just the smarter option, albeit the more difficult one, because it provides us with both the efficiency and versatility that we needed in order to meet all of our requirements.
    - In the hardware department, the option did exist for our team to use the incredibly low cost board, but the decision was made to favor versatility once again. The PlutoSDR board has a more narrow frequency band and leaves both our customer and our group no room to expand the scope of this project. In order to expand our project to body imaging and tracking, we would need a frequency range that goes up to at least 4GHz. Unfortunately, the PlutoSDR only goes up to about 3.8GHz. PlutoSDR is an excellent board and almost meets the bare requirements for our project, but it is far too shortsighted of an option when considering future plans that our customer may have for this project. The Zedboard offers far more capabilities at a price that is already dramatically below our upper bound on price, making it the optimal choice that allowed us room in case any requirements changed.
- How was the work at each phase (requirements, design, implementation, testing & evaluation) verified against the outcomes or prior phases?
    - At the end of the requirements phase, we made sure that each need was well-documented in order to reduce confusion later on. We also compared the requirements to our goals, making sure that the specifications properly produce a system that does what we need it to do and made sure that the requirements laid out a path of specificity leading to our final goal.
    - At the end of the design phase, we determined how we planned to implement different requirements we had to achieve. We made decisions on which software and hardware options we wanted to pursue by weighing the pros and cons of each and made a design as a group and through consulting our customer the research assistant he assigned to help us on what they thought the best plan of action would be.

  - At the end of the implementation phase, we compared the implementation with our original requirements, making sure that all specifications had been fulfilled. We made sure that we had a system that was in line with what our customer desired from us.
  - At the end of the testing and evaluation phase, we made sure that all system components were working together properly and that our system as a whole had a level of accuracy that met our requirements and expectations and also that it would consistently work as intended, with little error.
  - At the end of each phase, we made sure that the results of our findings and conclusions were still in line with our initial goal. Additionally, our reports were read and verified by the customer (Dr. Fathy). Finally, throughout the whole process, we collaborated with Dr. Fathy's graduate research assistant, Farhan Quaiyum who has been working on this concept for some time and has been extremely helpful.
- Describe the results and/or outcomes of the project
  - Our final design is able to read vital signs of a subject who is sitting within range and displays a continuously updating window that averages the most recent 10 seconds of data and updates every 1.25 seconds. It is able to output a frequency within 5% accuracy when scanning an actuator that moves at a fixed rate square wave, and to the best of our testing capabilities is able to consistently and accurately display a patient's respiratory rate. If the patient holds his or her breath the system can consistently and accurately display heart-rate, if the patient is breathing then the heart rate readout is significantly less dependable, but it still does sometimes work. We did not implement stepped frequency technologies because the graduate student who has been working on this problem was having troubles getting it to work on the system currently in place in the lab and was not able to eradicate all of the problems with enough time for us to implement the technology into our system. Because this was effectively out of our control and because we were in frequent communication with our customer and he agreed that it was not a reasonable task by the end of the semester, we do not feel like we failed this task. Overall, we feel that we were successful in designing a low cost, portable system that is able to quickly and accurately measure and display vital sign information for a patient in question.

Lessons Learned:

There were many lessons that were learned throughout the project this semester that will make everyone in the group better engineers and overall professionals in the careers we pursue later in life. One of the first lessons we learned about in engineering and the engineering design process is how much of the engineering design process comes outside of actually designing, building or testing of a product. There is a lot of work to be done before an engineer can even begin designing a solution, especially when working with a team, and even more so when that team is multidisciplinary. Before anything can be done the team must first organize themselves and come to a firm and clear understanding of roles that each member has and the responsibilities for which each member is accountable. This is vital in order for the group to function as a cohesive unit, or else there can be errors and other problems caused by miscommunications and misunderstandings. Our group learned this both in class during lectures and experimentally. Though this is a multidisciplinary project, much of the work was with computer programming and dealt with knowledge of FPGA's. The project thus depended heavily on the knowledge of our computer science and computer engineering majors, and left the electrical engineers somewhat stranded for awhile. We had sit down and organize ourselves based on our individual strengths in order to complement each other's work to yield the best solutions.

Another lesson our team learned is how important it is to narrow our scope and focus on a few problems at once rather than broaden our scope when there are still unsolved problems on the table. We ran into many situations where one of our first goals would be left unsolved, yet we would still continue to add on new goals. Our original plan was to implement SFCW within the first month of the design process. This turned out to be tricky, so our focus was taken away from the problem over time while new goals were added on top of it. Eventually, SFCW was scrapped because we could not figure out the problem in time. This was disappointing for us, because this was a very important goal that would have opened up many doors to new applications for our project.

In future projects, we would like to see the team put more focus on implementing SFCW with the SDR. We believe that this would take the project to the next level. CW radar only lets us read the vital signs of one person and cannot give us distance information. With SFCW, we would not only be able to read vital signs, but we would also be able to gauge the distance of the patient, track the patient, and even detect the vital signs of multiple patients.

Relevant coursework is evident on both sides of the electrical engineering and computer science spectrum. On the electrical engineering side, signals and systems is a key course for this project. In order to extract the respiratory rate and heart rate of the patient, signal processing must be applied to the transmitted and received signals. It is also beneficial to have a general knowledge of wave propagation, the doppler effect, and antennas. In a mix between electrical engineering and computer science, it was also important to know about Software Defined Radio. Fortunately, a new course at the university has just been started that teaches the basics of SDR as part of the coursework. On the computer science side, obviously any class dealing with C programming or Matlab programming was extremely beneficial. At first, the entire project was based off of Matlab code. As the project progressed, all of this code was transformed into C code that is loaded onto the board. Without this relevant coursework, we simply would not be able to tackle this project.

Team Member Contributions:

James Bates: James Bates is an electrical engineering major and became the team leader in the group when he was appointed by his group to email the group members to the T.A. His primary tasks involved much of the administrative work associated with keeping track of the the team and all of the tasks for the team throughout the semester. The largest of his tasks was to have a constant understanding of the progress on the project and to report that progress to the customer in weekly meetings. James also worked to make sure that due dates were met, tasks were assigned and clearly understood by the end of team meetings and he made the executive decision when there was an executive decision that needed to be made. James also brought a solid understanding of high frequency circuitry and radio transmission to the group to help the group to have a solid fundamental understanding of the basis of the project and he assisted in writing papers, preparing slides and presenting in class.

Steven Engel: Steven Engel is a computer engineering major, and focused largely on the low level software aspects as well as the marriage between software and hardware. Most of Steven's work was done with James (Tucker) regarding software selection and implementation. A smaller component of Steven's work involved working with James (Bates), Chandler, and Fangzhou by commenting on hardware and how the hardware selection would affect the overall system based off the compatibility with different software options. Steven brings embedded systems programming and development experience from computer engineering specific classes and a summer internship. Non-technical roles including writing papers and slides, in addition to presenting in class.

Chandler Bauder: Chandler Bauder is an electrical engineering major who worked on both the computer science and the electrical engineering parts of the assignment. Chandler was the first to have any experience with the system due to his work with Dr. Fathy over the summer. Chandler worked closely with Steven when transferring the signal processing code from MATLAB onto the Raspberry Pi and when deciding how to most effectively capture the vital sign data with the Zedboard code. Along with assisting with the code writing and testing the system, he also worked closely with Farhan Quaiyum, one of Dr. Fathy's graduate students, to try to figure out SFCW radar, explore alternate signal processing techniques, and discuss improvements to the system. Chandler also helped to write presentations and reports.

James Tucker: As the only computer science major of the team, James has focused primarily on the software implementations and preparing much of the code for the Zedboard and Raspberry Pi. James has reviewed the previous research for signal processing and collaborated with the team to decide on the best software implementation. His experience with quality assurance has helped forecast possible failures in solution strategies, as well as diagnose the solutions to specific project requirements and troubleshoot when there were issues in the code. Many of the documented requirements and specifications were outlined by James. James has also helped organize and delegate work for the presentations for the project requirements and solution strategies.

Fangzhou Liu: Fangzhou Liu is an electrical engineering major who primarily assisted with preparing presentations, writing papers and presenting to the class. He came in at a slight disadvantage due to the fact that he joined the group late after being assigned to it by one of the teaching assistants. He is also not a native english speaker, so there were a number of challenges to for him to overcome quickly at the beginning of the semester. There were a number of times where it was difficult to coordinate with him and a series of miscommunications due to a partial language barrier in the first semester but after he quickly learned how to help anyway he could and perform any tasks asked of him. Despite the hurdles he faced, he showed up to team meetings and communicated when he would not be able to make it to the best of his ability; he also helped to prepare presentations and present in front of the class.

Approval Signatures:

Dr. Fathy (Customer) A Fathy Date: 4/24/2018

James Bates Date: 4/24/18

Steven Engel Steven Engel Date: 4/24/18

Chandler Bauder Chandler Bauder Date: 4/24/2018

James Tucker James Tucker Date: 4/24/2018

Fangzhou Liu Fangzhou Liu Date: 4/24/2018