Utilizing Ultra-Wideband as a High-Speed Datalink in Medical Ultrasound Imaging

B. Groenen
b.groenen@student.tue.nl, ID 1352008

***Abstract*—This paper presents a study to assess the feasibility of Ultra-wideband (UWB) as a wireless communication link for ultrasound imaging at home. The aim is to design and realize a custom UWB link architecture for a high data rate transmitter that interfaces with multiple Analog-to-Digital Converters (ADC) to power efficiently communicate data from an array of ultrasound transducers to a wireless receiver. These data will be stored on a host computer which will process them into usable images. In detail, the transmitter samples the ADC outputs using a Static Random Access Memory (SRAM) buffer coupled to a Microcontroller Unit (MCU) at 4 MHz for buffering data, and this MCU interfaces with the UWB transceiver at 1.25 MHz using a Serial Peripheral Interface (SPI) bus. The UWB transceiver sends the data wirelessly to another transceiver linked to a receiving computer using a Universal Asynchronous Receiver/Transmitter (UART) bus. UWB is chosen over other known protocols such as WiFi or Bluetooth due to its lower energy per bit and high data rate, which makes UWB especially suitable for low-power applications. Utilizing this concept architecture, a link of 866.832 kb/s is demonstrated where the transmitter consumes 0.844 W and the chosen transceiver chip consumes 2.23 mW.**

I. Introduction

Ultrasound imaging is a crucial part of modern healthcare. Despite the availability of many other diagnostic imaging techniques, ultrasound is widely used as it does not involve ionizing radiation, is non-invasive, cost-effective, and widely available [1]. Consequently, the technique is often used in multiple medical disciplines. For instance, ultrasound is used to examine organs like the heart or a fetus in the uterus of a pregnant woman with echography [2], it is used to measure blood flow using Doppler ultrasound [2], and it is used to measure the stiffness of organs with elastography [3] next to serving many other diagnostic purposes.

Still, there are multiple disadvantages to ultrasound diagnostic techniques. Firstly, this procedure currently needs expert operators who know where to point the probe. Secondly, due to aging of the population, doctors and hospital staff have to do more work with fewer people. As a result, precious hospital resources have to be allocated to imaging instead of treating patients. Lastly, the current technology is bulky and expensive, which makes the technology unsuitable for use at home, meaning that images must be made in hospitals.

To address these issues, portable ultrasound systems could be provided to patients for making these images at home. This helps save precious hospital resources. Consequently, the transducers provided to the patients need to cover large areas [4] and be intelligently controlled to minimize the need for professional interventions, while ensuring that the device remains portable without affecting the quality of the diagnosis. To achieve this, a compact battery would be necessary, which makes the device's power consumption highly important. Additionally, real-time ultrasound imaging needs a high bandwidth. This means that an efficient and fast wireless link is required for a high data rate and low-power setup. Ultra-wideband (UWB) technology could be interesting as a wireless high-speed data link in portable wireless ultrasound systems.

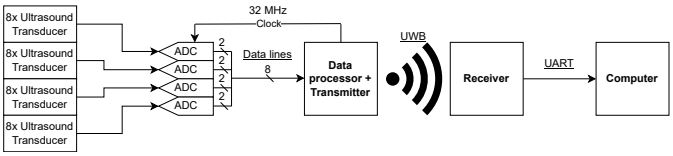

Fig. 1. Full overview of the ultrasound system

UWB is characterized by many key features such as its lower power consumption per bit compared to well-known alternatives such as WiFi or Bluetooth [5] and its large channel bandwidth of more than 500 MHz [5]. Due to its large channel bandwidth, UWB has very short pulses in the time domain (nano- or picoseconds) [6]. According to the Shannon-Hartley theorem, less power is needed for a target channel capacity with a large bandwidth than with a small bandwidth, which is a reason why UWB shows promise. Moreover, the technology features a large immunity to interference due to its large frequency band [6]. This makes UWB an excellent candidate for low-power communication systems.

This paper presents a feasibility study of whether UWB is suitable for ultrasound imaging at home along with the design and realization of a concept UWB architecture consisting of both a transmitter and receiver capable of sending and receiving image data respectively, as seen in Fig. 1. Section II provides a summary of the chosen UWB transceiver, while section III delves deeper into its limitations. Section IV provides insight into the custom UWB link architecture, whereas one of its key elements, the sampler, is presented in Section V. In Section VI, the methodologies used for experimental characterization are illustrated. The characterization results are presented in Section VII followed by a discussion in Section VIII and a conclusion in Section IX.

M.W. Timmermans, E. Cantatore

II. Ultrawideband Transceiver Summary

UWB operates in the 3.1 to 10.6 GHz frequency band [5] and has a large channel bandwidth starting from 500 MHz in which it can send signals in the form of very short pico- or nanosecond pulses [6]. Utilizing short pulses in the time domain saves transmission power over time for a target channel capacity, which in turn increases the efficiency of the transmission.

In this particular paper, the off-the-shelf SR1020AB-EVK1P4-MP110 Evaluation Kit (EVK) from SPARK Microsystems is utilized. It contains two boards with an STM32G473RET6 Microcontroller Unit (MCU) from ST-Microelectronics along with an antenna assembly containing an SR1020 UWB transceiver Integrated Circuit (IC) [7]. The SR1020 operates within the 6 to 9.3 GHz frequency band and is capable of providing a bit rate up to 20.48 Mb/s [8]. This UWB transceiver currently has the highest commercially available bit rate, which is why is it chosen. Next to this, the EVK features an expansion connector which has additional Input/Output (IO) ports that the MCU can use to read or write data [7].

III. Limitations of the UWB Transceiver

A. Sampling Speed

Multiple Analog-to-Digital Converters (ADC) are connected to ultrasound transducers as shown in Fig. 1. The data from these ADCs are sent over an 8-bit parallel interface. Additionally, these ADCs require that the data have to be sampled at a frequency of 32 MHz at the falling edge of each clock cycle. Despite the MCU of the EVK having a maximum clock frequency of 170 MHz [9], it cannot sample all the data at 32 MSample/s without data loss, as the operation of reading the General Purpose Input/Output (GPIO) bus and storing the data in memory needs 40 clock cycles. Consequently, the maximum sampling frequency at which the MCU can sample without data loss is about 4 MSample/s.

B. Data Sampling

The STM32G4 family of MCUs has multiple GPIO buses, each with 16 IO ports [9]. When sampling, either the lower 8 IO or the upper 8 IO ports of one specific GPIO bus have to be read out simultaneously. The MCU cannot read IO from different buses at the same time, this has to be done in different clock cycles [10].

Even though the expansion connector on the EVK hosts enough available IO ports, these ports are spread among multiple GPIO buses. Consequently, efficient sampling is impossible which leads to a lower sample rate than 4 MSample/s which is why another solution should be used.

The logic level of the ADCs in Fig. 1 is 1.2 V, which is below the threshold voltage for the MCU [9]. Due to this, the MCU is unable to sample the incoming data at this amplitude.

C. Memory limitations

There are a total of 32 ultrasound transducers of which 8 are connected to one ADC, resulting in a total of 4 ADCs as depicted in Fig. 1. Per ADC, 2 output data lines are provided, totaling 8 data lines. Furthermore, every ADC has to be sampled at 32 MHz, sending bursts of data for 250 µs, which leads to a total size of 8000 bytes per data frame. Moreover, the transducers will measure ultrasound waves at 100 different angles to create an image, resulting in an image size of 100 frames or 800 kB. The maximum available memory on the MCU is 128 kB [9], which means that an entire image does not fit into the memory of the MCU.

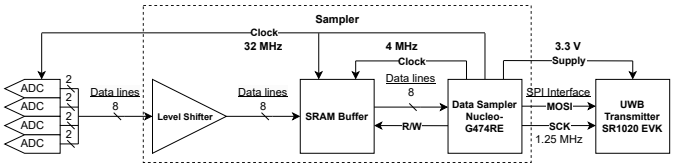

Fig. 2. Implementation of the UWB transmitter in an ultrasound application

IV. Custom UWB link architecture

A. Transmitter

Since the EVK does not have a lower or upper half of a complete GPIO bus available on the expansion connector, another host for sampling the data must be chosen. This sampler is discussed in Section V. Furthermore, the EVK is modified so it has an additional Serial Peripheral Interface (SPI) bus available and can be powered using an external supply. The full architecture of the custom UWB transmitter is thus shown in Fig. 2.

B. Receiver

The receiver is an unmodified EVK. It features an easy-to-access Universal Asynchronous Receiver and Transmitter (UART) bus over which a terminal on a host computer can be opened for data reception. Furthermore, the EVK initializes the SR1020 to be a receiver and gathers data until one full data frame is received. This data frame is then sent to the UART terminal whilst another data buffer is received. Once the first frame is sent and the second data frame is filled, the MCU sends the second data frame and refills the first data frame, until all data are sent. Once all data are transmitted, the host computer will create a black-and-white heatmap representation to construct ultrasound images using the transmitted data frames.

V. Sampler (Extension)

The standalone sampler implementation utilizes the ST-Microelectronics Nucleo G474RE development board, hereafter referred to as the Nucleo. This board features the STM32G474RET6 MCU which is from the same family as the MCU on the EVK. The only difference between this MCU and the one on the EVK is that the STM32G474RET6 features high-resolution timers [11]. Additionally, the Nucleo has multiple full GPIO buses and timers available. Timers are utilized to generate timed events such as interrupt requests and external clocks. Since it is from the same family as the EVK MCU, the MCU's maximum sample rate is 4 MSample/s.

Consequently, this means that the MCU does not directly meet the specifications for sampling the ADCs at the requested 32 MHz.

A. Sampling

A Static Random Access Memory (SRAM) buffer is inserted between the ADCs and the Nucleo as visible in Fig. 2. Specifically, the IS61WV5128FBLL-10TLI asynchronous SRAM buffer from ISSI is chosen. This SRAM buffer samples and stores an entire data frame at 32 MHz, which the Nucleo will later read at 4 MHz. Moreover, the chosen SRAM buffer supports write speeds of 125 MHz and can output 3.3 V logic levels [12]. Along with the 8-bit parallel interface, this leads to a theoretical bitrate of 32 Mb/s, which is higher than the throughput of the SR1020. Therefore, it does not lead to a bottleneck making the chosen SRAM buffer suitable for this purpose.

Since the logic level of the ADC is 1.2 V, it is below the threshold voltage of the SRAM buffer [12]. For this reason, a level shifter is inserted between the ADC and the SRAM buffer to boost the logic level to 3.3 V. This enables the ADC to interface with the SRAM buffer. The SRAM buffer and level shifter are not validated in this paper.

B. Clocking

The ADC and the SRAM buffer both require external clocks to operate and are required to be in phase with each other. One clock source is used for the sampling architecture to ensure phase synchronization across the entire implementation. The clock source chosen is the internal 16 MHz RC oscillator of the Nucleo development board. Furthermore, this 16 MHz clock frequency is multiplied by a Phase-Locked Loop (PLL) to be 160 MHz and supplies the system clock on which the MCU of the Nucleo operates [9].

The clock frequency of 160 MHz is specifically chosen to be an integer-scaled version of the required clock frequencies for both the ADCs and the SRAM buffer. Furthermore, the Nucleo handles the clocking of the ADC by providing it with a 32 MHz square wave created by one of its internal timers. Additionally, this 32 MHz clock is shared with the SRAM buffer during a read event of the ADCs. After this event, the data are sampled by the Nucleo which disables the 32 MHz clock signal and now provides it with a 4 MHz clock signal. This lower clock frequency is generated by the system clock of 160 MHz scaled by a factor of 40 by a timer, creating a 4 MHz square wave.

C. Data transmission and memory management

The Nucleo communicates with the transmitter EVK using a SPI bus at 1.25 MHz. This leads to a maximum data rate of 1.25 Mb/s, a bottleneck for the entire transmitter system. Furthermore, 1.25 MHz is the limit at which data is reliably sent to the transmitter EVK.

As discussed in Section III.C, the available memory of the Nucleo is insufficient for storing an entire ultrasound image. therefore, two data buffers of 8000 bytes are utilized to sample

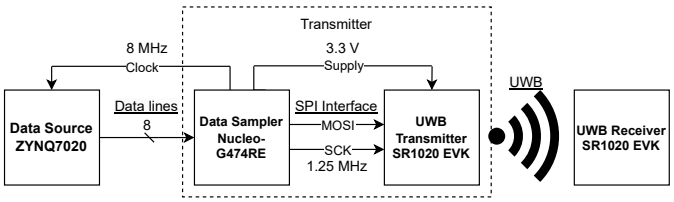

Fig. 3. Validation setup for the custom UWB architecture

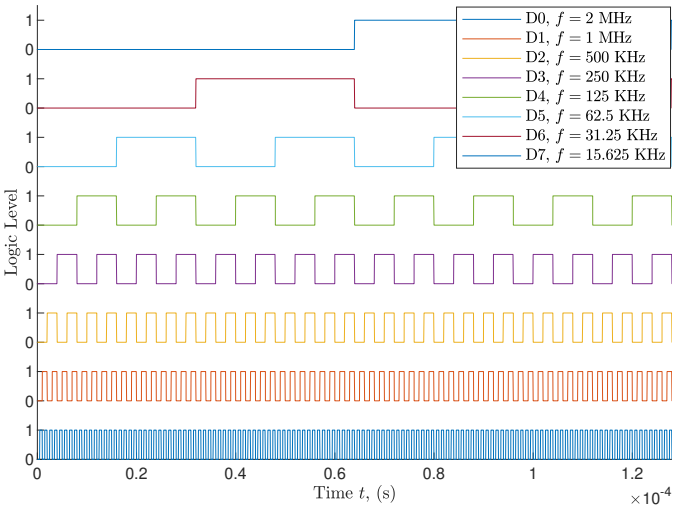

Fig. 4. Measured periodic data from the FPGA to be sampled by the MCU

data frames sequentially. When one data buffer is filled, the MCU sends the contents of this data buffer over SPI to the transmitter EVK while it simultaneously samples data into the second data buffer. Once the first data buffer has been transmitted and the second buffer is filled, the MCU sends the second data buffer over SPI and begins sampling new data into the first data buffer. This will continue until one image is sent. This way, only 16 kB of the Nucleo's SRAM is used to send one entire image.

VI. Custom UWB link validation methods

To ensure that the custom UWB link concept works, it is tested using a Field Programmable Gate Array (FPGA) which produces a known signal counting from 0 to 255 binary over an 8-bit parallel bus at 4 MHz. The FPGA chosen for this purpose is the AMD ZYNQ XC7Z020-1CLG400C that is mounted on a TUL PYNQ Z2 development board. This setup is depicted in Fig. 3 and the data signals sent to the MCU with a clock frequency of 8 MHz are depicted in Fig. 4. With the setup shown in Fig. 3, possible errors in the sampling, processing, and wireless link can be evaluated. Furthermore, the setup in Fig. 3 is used for obtaining the results reported in this paper.

A. Power consumption

To measure the current consumption of the transmitter, the receiver and the SR1020 antenna assembly respectively, a shunt resistance is put in series between the power source and

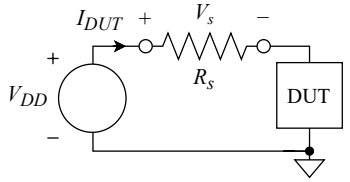

Fig. 5. Power measurement setup of all measured devices

TABLE I
TEST PARAMETERS TRANSMITTER

Parameter	Value	Unit	Description
$V_{DD,Nucleo}$	5.06	V	Supply voltage
R_s	0.210	Ω	Shunt resistance

the Device Under Test (DUT) as seen in Fig. 5. With this, the DUT power consumption can be approximately calculated as

$$P \approx V_{DD} \cdot I_{DUT}, \tag{1}$$

where P is the DUT power consumption, V_{DD} is the supply voltage of the DUT and I_{DUT} is the current consumption of the DUT. (1) provides a good approximation as long as the power dissipated in the shunt resistor is much smaller than the power dissipated in the DUT.

1) Transmitter: The transmitter consists of both the Nucleo and the transmitter EVK as seen in Fig. 2. Additionally, the test setup parameters for the transmitter are listed in Table I.

As the entire transmitter is considered, the supply voltage input of the transmitter EVK is connected to the 3.3 V output of the Nucleo to measure the total current draw of the transmitter implementation. To measure the current consumption, the voltage drop across the shunt resistance R_s is measured using an oscilloscope to account for transient behavior. The combined current consumption of the assembly is calculated as

$$I_{DUT} = \frac{V_s}{R_s}, \tag{2}$$

in which I_{DUT} is the current consumed by the DUT, V_s is the voltage drop across R_s, and R_s is the shunt resistance.

2) Receiver: As with the transmitter, the values of the shunt resistance and supply voltage for the measurement setup are listed in Table II. The shunt resistance has a higher value to make a more accurate current measurement due to the lower current draw of the EVK only. Furthermore, the voltage drop across the shunt resistance is measured using an oscilloscope to calculate the transient current behavior with (2).

3) SR1020 IC: The current consumption of the SR1020 transceiver IC is calculated with (2). The voltage drop across the shunt resistance is measured with an oscilloscope to account for transient behavior. Furthermore, the power is calculated with (1). The parameters used are listed in Table II, as the SR1020 IC uses the same supply parameters as the EVK.

TABLE II
TEST PARAMETERS EVK AND SR1020 ASSEMBLY

Parameter	Value	Unit	Description
$V_{DD,EVK}$	3.30	V	Supply voltage
R_s	1.18	Ω	Shunt resistance

Fig. 6. A black-and-white image reconstructed after reception over UWB

B. Sampling

The Nucleo is connected to the FPGA data source as shown in Fig. 3 and samples 100 data frames of 8000 bytes at 4 MSample/s. For testing purposes, the data of the last two sampled data frames are sent over a UART terminal hosted by the Nucleo to be read out by a computer. The last two data frames are read out specifically, as it does not introduce a delay during sampling. Otherwise, the Nucleo has to copy the chosen sampled frame to another buffer, which introduces delay during sampling as it cannot be overwritten during the copying process. To verify that sampling happens without errors, a plot can be shown with the first 256 bytes of these two data frames along with its transmission over SPI at 1.25 MHz to ensure that the correct data are sent over to the transmitter EVK.

C. Data rate

The data rate of the link between the two EVKs is measured utilizing a Python script that counts the number of incoming bytes on the receiver's UART bus. However, the measurement period is set to 30 seconds, to get a better average of the bit rate. This bit rate is calculated with

$$R_b = \frac{n}{30} \cdot 8, \tag{3}$$

where R_b is the bit rate and n is the total number of bytes received over a period of 30 seconds.

D. Data integrity

A 256x256 black and white image equaling 65536 bytes is generated from an ultrasound image and is shown in Fig. 6. This image is stored in the memory of the Nucleo, to be sent to the transmitter EVK over SPI. This data is in turn transmitted over UWB to the receiver end and converted to an image by a Python script.

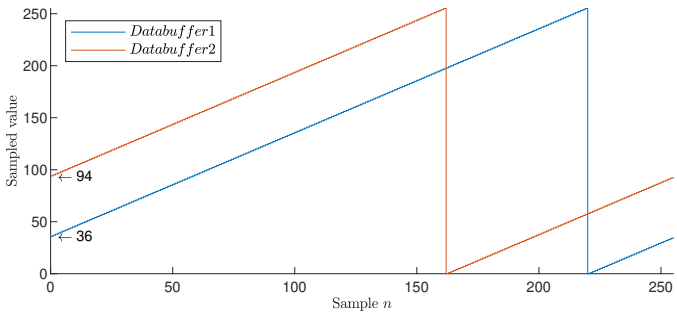

Fig. 7. Visualised UART transmission of the sampled data buffer contents in the form of a plot

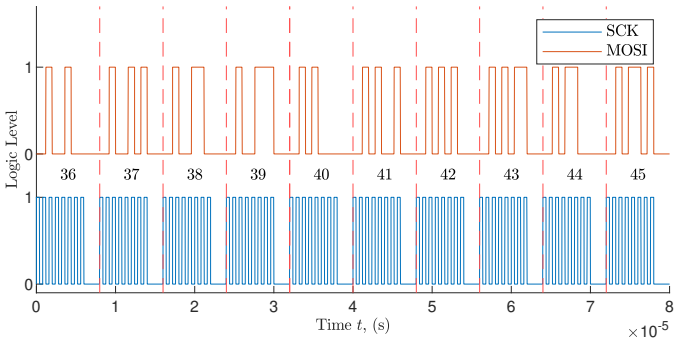

Fig. 8. Measured beginning of the first data buffer SPI transmission

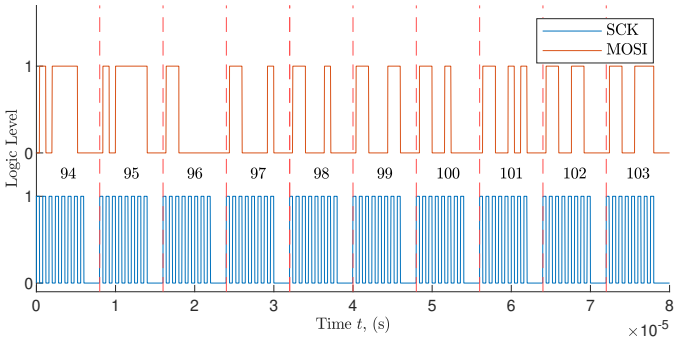

Fig. 9. Measured beginning of the second data buffer SPI transmission

E. Total system

In order to verify the total system functionality, data from the FPGA counting from 0 to 255 is sampled, sent over UWB, received, and transferred to a computer. These data are then visualized in the form of a 32x256 grayscale image, or 8192 bytes. This is slightly larger than one data frame. What should be visible is a continuous gradient of 256 possible gray values. Additionally, the FPGA outputs data continuously, so sampling starts anywhere in the range of 0-255.

VII. Custom UWB link results

A. Power consumption

1) Transmitter: A voltage drop of 35 mV is found across the shunt resistance along with basically no transient behavior. Consequently, the current consumption calculated with (2) is found to be 0.167 A. Using (1) and the supply voltage listed in Table I, this current consumption leads to a transmitter power consumption of 0.844 W.

2) Receiver: There is a measured voltage drop of 98 mV across the shunt resistor of 1.18 Ω where no transients occurred. Using (2), a current consumption of 0.083 A is calculated when idling, sending, and receiving data. This corresponds to a power draw of 0.274 W when calculating it with (1) using the supply voltage mentioned in Table II.

3) SR1020 IC: When initialized by the MCU of the EVK, a voltage drop where no transients are visible is measured of 0.5 mV across the shunt resistance. Using (2) and the parameters listed in Table II, this corresponds to a current consumption of 0.45 mA. During transmission and reception, the voltage drop measured is 0.8 mV. Additionally, no transients are present. This voltage drop corresponds to a current draw of 0.68 mA. As a result, a power consumption of 1.48 mW is found during idle, whereas the power consumption during activity is found to be 2.23 mW.

B. Sampling

The first 256 sampled bytes of the last two data buffers sampled by the Nucleo and sent out over UART are shown graphically in Fig. 7. This figure shows an increasing value without double or missing data in both of the data buffers, which demonstrates that data are sampled correctly at 4 MSample/s.

Time diagrams depicting the first few bytes of the data transmission over SPI to the transmitter EVK containing data buffers 1 and 2 are shown in Fig. 8 and Fig. 9 respectively. These figures, generated with data acquired from an oscilloscope, show that the data in the data buffers are correctly sent over SPI towards the transmitter EVK.

C. Data rate

A maximum payload of 124 bytes can be transmitted per transfer to the receiver [7]. The received data over a period of 30 seconds is 3,250,620 bytes. This results in a bitrate of 866.832 kb/sec utilizing (3), or slightly more than 1 image per 8 seconds.

D. Data Integrity

A 256x256 black and white image is sent to the transmitter EVK by the Nucleo over SPI, which is then sent over UWB, received by the receiver EVK and reconstructed as shown in Fig. 6. This figure shows that a UWB link is established and that the custom architecture can reliably send image data wirelessly. Furthermore, a comparison between transmitted and received data shows no differences, indicating that the image is transmitted and received without bit errors.

E. Total system

As seen in Fig. 10, a gradient starts halfway through the image. These are the data sampled from the FPGA counting from 0 to 255, where 0 is represented with black, and higher values are represented with a lighter gray until 255, which is represented with white. Data in this plot are presented from left to right with a width of 256 pixels, where each pixel represents

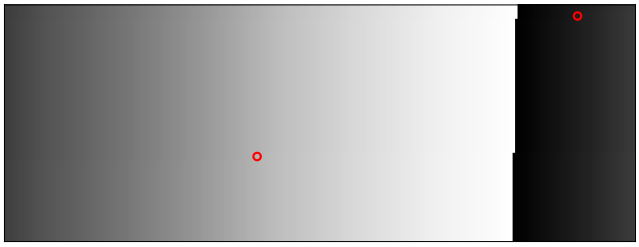

Fig. 10. Sampled data from the FPGA as an image after reception over UWB

a sample. With larger samples, double-clocking is likely to occur on the FPGA. Any double-clocking events occurring in this frame are marked by the red circles shown in Fig. 10. A clear gradient visible in the figure shows that the transmitter can correctly sample and send frames and that the receiver EVK can receive said data without any artifacts being created anywhere in the system apart from the double-clocking events. Out of the total samples, 0.0244 % are double-clocked. The reason for double-clocking occurring is discussed in Section VIII.

VIII. Discussion

A. Architecture

Although the custom UWB link architecture is able to sample at 4 MSample/s, transmit data, and receive said data, it is important to highlight the things that did not work as intended.

Since power minimization is important and the EVK is unable to sample efficiently, the Nucleo-based custom sampler was used to drive both the SR1020 transceiver IC next to hosting the sampling process. For this reason, the MCU on the EVK was removed. This would mean that instead of two STM32G4 MCUs, only one MCU would be powered, decreasing power usage to that only of the Nucleo and the SR1020 IC.

Driving the SR1020 IC with the Nucleo did not work, because the Software Development Kit (SDK) was unable to initiate a link between the two. Additionally, all the internal timers of the Nucleo's MCU were allocated to the SR1020 IC for its operation. Two of these timers were necessary for generating the clock signals for the SRAM buffer and the ADCs, which made this setup unable to work.

For this reason, the MCU of the EVK was reassembled and the SPI bus on the EVK expansion connector was enabled to handle incoming data from the Nucleo to the EVK.

The data rate of the UWB link could be improved by altering the code to reduce the EVK MCU's processing between data transmissions, as this costs clock cycles and restricts transmission speed. Additionally, the data transmissions are limited by the SDK to 124 bytes [7], which further limits the data rate.

B. Nucleo

There was potential for the Nucleo to operate at a higher sample rate with the use of Direct Memory Access (DMA) [10], as the STM32G4 series supports this feature. This would have eliminated the clock cycles needed to store data in memory from the sampling registers, which would result in quicker sampling and thus a higher sample rate of up to 8 MSample/s. However, no working prototype using DMA could be established. Consequently, the idea of DMA was abandoned to prioritize creating a working prototype. Furthermore, the SPI bus was unable to operate at the intended speed of 40 MHz due to crosstalk, length mismatches and mismatched termination impedances. Consequently, the SPI bus operated at 1.25 MHz at which the transmitter EVK was able to receive data reliably. With shorter cables and an impedance matched to the input pin, this frequency would likely improve.

C. FPGA

The termination impedance between the Nucleo and the FPGA was not properly matched, resulting in an overshoot on the FPGA clock signal subsequently causing double-clocking at the FPGA. This made the counter skip one digit which led to jumps in values between samples. This double-clocking behavior is visible in Fig. 10.

IX. Conclusion

The custom UWB link architecture has demonstrated its functionality by achieving a bitrate of 866.832 kb/s and successfully transmitting images wirelessly. Furthermore, the transmitter architecture consumes 0.844 W while the SR1020 IC consumes 2.23 mW during operation. This indicates that UWB is a reliable and low-power data link feasible for medical ultrasound. However, the interface with the data source needs power consumption optimization to be feasible. Additionally, further research is needed to improve the data rate, as it is currently not high-speed.

References

[1] C. F. Dietrich, C. B. Sirlin, M. O'Boyle, Y. Dong, and C. Jenssen, "Editorial on the current role of ultrasound," 9 2019.

[2] A. Carovac, F. Smajlovic, and D. Junuzovic, "Application of ultrasound in medicine," *Acta Informatica Medica*, vol. 19, p. 168, 2011.

[3] J. L. Gennisson, T. Deffieux, M. Fink, and M. Tanter, "Ultrasound elastography: Principles and techniques," pp. 487–495, 2013.

[4] P. L. van Neer, L. C. Peters, R. G. Verbeek, B. Peeters, G. de Haas, L. Hörchens, L. Fillinger, T. Schrama, E. J. Merks-Swolfs, K. Gijsbertse, A. E. Saris, M. Mozaffarzadeh, J. M. Menssen, C. L. de Korte, J. L. P. van der Steen, A. W. Volker, and G. H. Gelinck, "Flexible large-area ultrasound arrays for medical applications made using embossed polymer structures," *Nature Communications*, vol. 15, 12 2024.

[5] W.-F. J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols."

[6] D. Coppens, A. Shahid, S. Lemey, B. V. Herbruggen, C. Marshall, and E. D. Poorter, "An overview of uwb standards and organizations (ieee 802.15.4, fira, apple): Interoperability aspects and future research directions," pp. 70 219–70 241, 2022.

[7] SPARKMicrosystems, "Spark sdk documentation." [Online]. Available: https://www.sparkmicro.com/sdk-docs/

[8] SparkMicrosystems, "Sr1020 datasheet."

[9] STMicroelectronics, "Stm32g474xb stm32g474xc stm32g474xe arm ® cortex ®-m4 32-bit mcu+fpu datasheet," 2021. [Online]. Available: www.st.com

[10] ——, "An4666 application note parallel synchronous transmission using gpio and dma introduction," 2016. [Online]. Available: www.st.com

[11] ——, "Stm32g4 series." [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32g4-series.html

[12] ISSI, "512kx8 high speed aynchronous cmos static ram key features," 2018. [Online]. Available: www.issi.com