Simple Open-Source UHF RFID Tag Platform

Nicolas Barbot
Univ. Grenoble Alpes, Grenoble INP, LCIS,
F-26000 Valence, France.
nicolas.barbot@lcis.grenoble-inp.fr

Pavel Nikitin
Impinj, Inc
Seattle, WA, USA
nikitin@ieee.org

***Abstract*—In this work, we present a simple open-source software-defined based UHF (RAIN) RFID tag that can be used for academic research. This work is a follow-up to the open-source reader work presented in [1]. The hardware associated with this tag is only composed of an envelope detector and an RF switch and uses a modular design. All operations related to the RFID protocol, which includes clock recovery, data recovery and frame synchronization are entirely realized in software and can be processed by an Arduino Uno platform. The purpose of this work is to encourage researchers and students to experiment with RAIN RFID technology, to understand its protocols and standards, and to improve the proposed tag design. All relevant files (including schematic and source code) will be released as open source to the community.**

***Index Terms*—EPC Gen2, Software Defined Radio, UHF RFID tag.**

I. Introduction

RADIO Frequency Identification (RFID) is based on the backscattering principle [2] and allows to drastically reduce the cost of a transceiver. The technology has been standardized in 2004 with GS1 EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Standard also known as ISO 18000-6c and updated to version 2 in 2018 [3]. Dozens of companies offer readers and tags compliant with the standard for all world regions. At this time, typical UHF tag composed of a single IC connected to a flexible antenna has a price between \$0.1 and \$1 depending on the quantity [4].

However, commercial tag chips do not offer a viable solution to study and understand the RFID protocol [3] since reader commands and tag replies exchanged on the air interface are generally not accessible to users. In a typical inventory round, users can simply access the EPC values of the tags detected by the reader and memory bank content. Note that all the operations and messages exchanged between the reader and the tags are actually hidden from the user which limits the understanding of the RFID protocol.

Moreover, a RFID protocol defines a rich set of parameters (for data link, modulation, estimated population, etc.) allowing the user to optimize the performance of the system to cover many different applications. However, these values are usually not available to the user or constrained during the chip design. For example, the slot counter generated by a tag after a Query command cannot be controlled since this value is entirely chosen by the pseudorandom number generator present inside each tag chip.

Software Defined Radio (SDR) is a recent paradigm allowing one to realize most of the operations of a transceiver in software. SDR designs offer a higher flexibility since most of the modifications can be simply realized by changing the software only. SDR architecture also allows educational perspective since the associated code can easily be understood and modified to realize and test new functions and cover new applications. The concept of a tag emulator and a discrete tag platform appeared since 2005 [5]. The most famous example is the WISP platform designed at University of Washington [6] The WISP is a fully passive discrete tag built around a MSP430 MCU and uses a fully discrete RF front end. Many other researchers have reproduced this architecture based on MSP430 [7]–[9]. Some other prototypes has been realized based on CPLD [10] and FPGA [11] to satisfy the strict timing constraints of the RFID protocol. Other interesting designs such as [12], [13] combine a MCU and an advanced UHF chip (*e.g*, Monza X Dura, SL900A, Rocky100, AS321X...). This architecture relaxes the hardware and protocol complexity but suffers from a lower flexibility. Finally, architecture based on USRP [14] as been proposed as sniffers but did include a backscatter modulator. Note that all these prototypes require the design of a custom RF PCB board and to need to handle complex firmware in VHDL or Assembly language. Moreover, most of the solutions (except [6] and [14]) do not provide source code of the firmware. The learning curve for students to reproduce those designs is quite steep.

In this paper, we present a simple low-cost SDR tag which is able to emulate the behavior of a UHF Gen2 tag. The proposed tag is theoretically able to process any command from a reader and to generate any reply defined by the standard in real-time. When receiving, this tag is able to extract the complete timing information associated to the reader command. During backscattering, each parameter of the reply can be set at any value (including values outside of protocol specification). Note that all tasks specific to the RFID protocol (including clock recovery, data recovery, frame detection...) are entirely defined in software and can be successfully processed by an Arduino platform. Moreover, this tag can be used to investigate the RFID commands sent by any given reader. The last point addressed by this paper is to encourage any researcher, student or person aiming to study the RFID technology to reproduce, use and improve the proposed design. For that, all hardware design files and software design are released under open source license to the community [15].

II. Architecture

A. Overview

The hardware of proposed RFID tag is extremely simple and represents less than 10 discrete components in total. The tag is composed of two distinct parts. The first is the envelope detector and data slicer used to receive reader commands. Design is identical as the one presented in [16] and can be implemented on generic protoboard.

The second one is the backscattering modulator used during tag replies. This modulator is composed of an RF switch connected to two different loads. Design has been largely inspired by [17]. Moreover, demodulator and backscattering modulator are separate components and are only connected through a 10 dB directional coupler. Note that a power divider or a circulator can be used instead of the coupler. An SMA connector allows one to connect a (single) external antenna but bistatic operation is also possible by using two different antennas. This modular design allows one to easily modify the tag architecture without requiring the design of a specific RF printed circuit board. Different possibilities are summarized in Fig. 1.

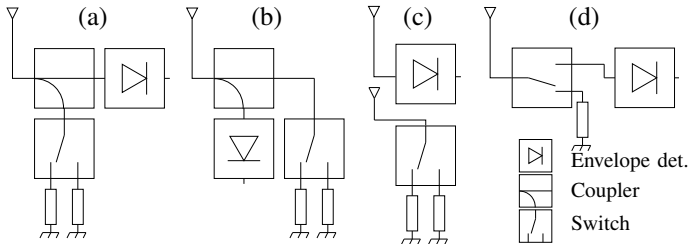

Fig. 1. Several possible tag architectures based on separate components. Envelope detector and switch are connected to the Arduino. Architecture type (a), (b) and (c) have been investigated in this paper.

Finally, both boards are directly controlled by the Arduino platform which executes the firmware implementing the RFID protocol. The complete schematic is presented in Fig. 2 and a picture of the tag type (b) appears in Fig 3. Note that the tag requires a power supply which can be provided by the USB cable or by a battery (and should be considered as a semi-passive tag).

The heart (or brain) of the SDR tag is an Arduino Uno platform [18] built around a AVR328p MCU (8 bits/16 MHz). This MCU can be considered as a low-performance MCU. However, this platform, due to its ease of use, have been used by thousands of users. Moreover, the presence of numerous libraries allows anyone to quickly add new functionalities easily. All the operations specific to the EPC Gen 2 protocol are actually handled, in real time, by the firmware executed by the MCU. This firmware represents the most important part of the design. As we will see, this simple design and the associated firmware allow one to emulate a UHF tag and build complex functionality above the RFID standard. The proposed solution also allows to realize some modes of operation that are not available in classical UHF tags.

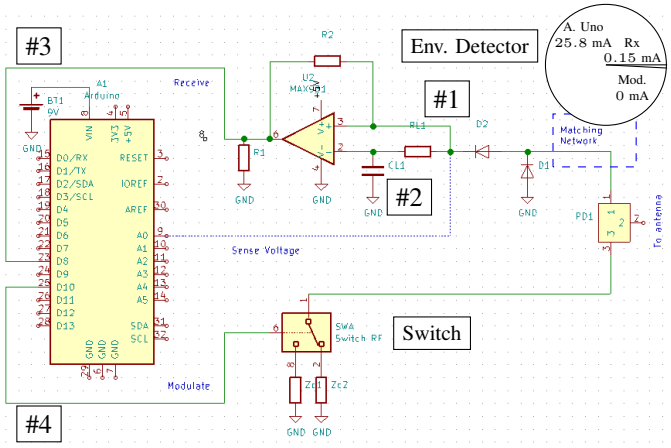

Fig. 2. Schematic of the hardware for proposed SDR RFID tag and current consumption by its various components. The upper part corresponds to the receiver based on the envelope detector and the data slicer, the lower part corresponds to backscattering modulator.

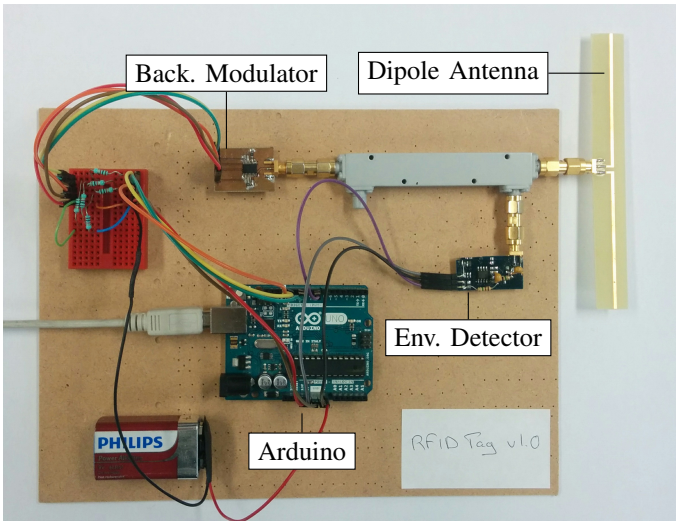

Fig. 3. Picture of the proposed SDR RFID tag [architecture (a), see Fig. 1]. The tag is composed of an envelope detector and a backscattering modulator both connected to the Arduino platform.

B. Receiver

The receiver architecture for the proposed tag is based on a 1-stage envelope detector to reduce the complexity. Identically to [16], the demodulator has been implemented in hardware and is based on a simple envelope detector and a data slicer (to dynamically set the amplitude of the decision threshold and produce a signal between 0 V and 5 V). The data slicer output is directly connected to a digital GPIO (in input) of the MCU (see Fig. 2). Fig. 4 presents the output of the envelope detector and the data slicer during the reception of a Query command. Data slicer output is equal to 5 V when the voltage of the envelope detector is higher than the amplitude threshold and equal to 0 otherwise. The threshold amplitude is obtained by low-pass filtering the output of the envelope detector. For the

TABLE I
EDGE DETECTOR AND BIT DECODER FOR A PIE WITH DATA-0 = 20 µS AND DATA-1 = 40 µS (THRESH_L = 240, THRESH_H = 260, PIVOT = 240)

t_i	2875	3129	3235	3431	4202	4400	2482	2704	3176	3379	3515	3705	3860	4038	4204	4372
Δt_i		254	106	196	771	198	63618	222	472	203	136	190	155	178	166	168
$\hat{b}_j$					1		1		1		0		0		0	

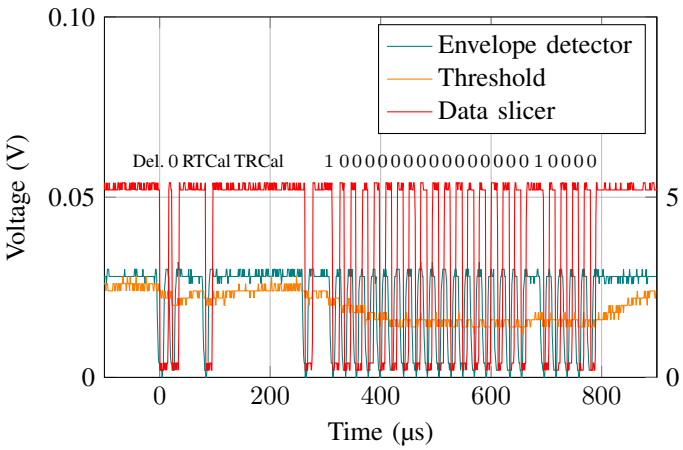

Fig. 4. Analog signals involved in the receiver (envelope detector at #1, threshold at #2 and data slicer at #3).

```
1  ISR(TIMER1_CAPT_vect)
2  {
3    static unsigned int time = 0, state = 0;
4
5    TCCR1B = TCCR1B ^ (1<<6);
6    timing[i_glob] = ICR1;
7    time = timing[i_glob] - timing[i_glob-1];
8    if (time > THRESH_L && time < THRESH_H) //Delimiter
9    {
10     goDel = 1;
11     delimiter = i_glob;
12   }
13   i_glob = i_glob + 1;
14   if (delimiter != 0)
15   {
16     if ((state%2)==1)
17     {
18       if (time < PIVOT)
19         answer[j_glob++] = 0;
20       if (time > PIVOT)
21         answer[j_glob++] = 1;
22     }
23     state++;
24   }
25 }
```

Fig. 5. Interrupt handler for the ATmega328p (Arduino). This code realizes at the same time, clock recovery, data recovery and frame detection.

envelope detector (and data slicer), the associated bandwidth have been estimated at 5 MHz which allows to receive all supported tari values [3, Section 6.3.1.3.3]. Moreover, the sensitivity of this detector (*i.e.*, the minimum modulated power which can be detected by envelope detector and data slicer) has been estimated at -23 dBm. Finally, our architecture also implies that clock recovery, data recovery and frame detection have to be done entirely in software by the MCU.

For the software part, the decoder has been implemented based on a simple edge detector *i.e.*, the detector only estimates the time at which transitions occur in the (digital) received signal. Note that, no matched filter or correlator is used during the decoding. Edge detection can be accurately realized using a simple timer (already present in all micro-controllers). When the tag is ready to receive a reader command, the tag passes in receive mode. In this mode, an interruption is triggered for each transition (from low to high or high to low) detected on the GPIO pin. At each interruption, the value of the timer t_i is captured and saved in the micro-controller memory.

PIE encoding [3, Section 6.3.1.3.2] and commands can be decoded using only the t_i values. To produce the binary sequence, the decoder simply computes the time interval $\Delta t_i = t_i - t_{i-1}$ between consecutive transitions. A 0-data is decoded if the duration of the symbol lower than the pivot and a 1-data is decoded otherwise. The pivot is determined dynamically from RTCal parameter present in the preamble of the command (*i.e.*, form the TRCal [3, Section 6.3.1.2.8] and DR [3, Table 6-9] parameters). The complete algorithm is presented in Fig. 5.

Fig. 4, 5 and Table I can be used jointly to understand the principle of the decoder. For each transition of the data slicer (see Fig. 4, the value t_i captured by the timer is saved (see Fig. 5, line 6). Table I line 1 presents the first values of t_i corresponding to the transitions in Fig. 4. The time interval Δt_i is then computed (see Fig. 5, line 7). Table I line 2 also presents the first values of Δt_i. Frame detection is realized according to the PIE preamble [3, Section 6.3.1.2.8]. If the current duration is close to 12.5 µs (*i.e.*, between THRESH_L and THRESH_H, see Fig. 5 line 8), then a delimiter is detected which indicates the beginning of a command. Bit decoding is also done inside the interrupt handler by comparing each (even) duration to the pivot. Short duration are decoded as 0-data (see Fig. 5, line 19) and long duration are decoded as 1-data (see Fig. 5, line 20). Note that this algorithm allows one to realize, on the fly, frame detection, clock recovery and data recovery. Final decoded sequence is presented in Table I line 3 and on top of the data slicer curve in Fig. 4 where we can recognize the 4 first bits `1000` of a Query command. The interrupt code shown in Fig. 5 represents the most important part of the firmware. As we will see, the execution time of this function directly determines the decoding rate which can be supported by the tag.

C. Backscattering Modulator

The transmitter is based on an RF switch [19] connected to two different loads. Loads are made with discrete components

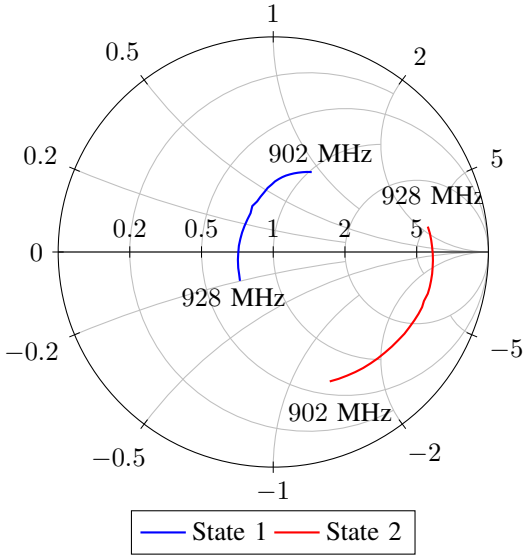

Fig. 6. Smith chart of the 2 impedance states between the switch and the dipole antenna in the bandwidth 902–928 MHz.

by using resistors and capacitors using the procedure described in [17]. Note that any other RF switch can be used for modulation. Fig. 6 presents the 2 impedance states between the switch and the dipole antenna. Note that the RF switch can easily support all BLF values of the RFID protocol [3, Section 6.3.1.3.3].

The API of the firmware has mainly been chosen at the bit level of the air interface. Thus, the programmer can generate any RN16 or EPC values. Helper functions allows the user to easily generate and check other required information (preamble, CRC bits...). Since the RF switch is directly controlled by a GPIO of the MCU, this design allows to generate any tag reply and to set every parameter defined by the protocol in software. Also, exact timing between tag replies can be accurately controlled and non-compliant replies can also be generated.

III. Performance Evaluation

A. Tag Thresholds

Tag performance can be characterized by the minimum power able to activate the tag, also called threshold POTF (Power on Tag Forward), and the effective backscattered power at the activation also called POTR (Power on Tag Reverse). These metrics can be easily compared to the performance of other tags and/or to the ARC specifications [20]. The POTF and POTR, in monostatic setup and free space propagation environment, are defined as:

$$\text{POTF} = \frac{P_{t\,min} G_r \lambda^2}{(4\pi)^2 d^2} \quad \text{POTR} = \frac{P_{r\,min} (4\pi)^2 d^2}{G_r \lambda^2} \qquad (1)$$

where λ is the wavelength, d is the distance to the tag and G_r is the reader gain. $P_{t\,min}$ and $P_{r\,min}$ are the minimal transmitted power required to activate the tag and minimal modulated power which can be detected by the reader respectively.

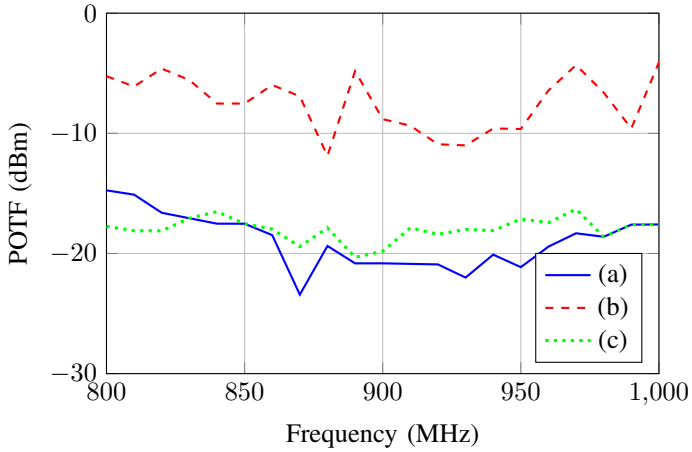

Fig. 7. Power On Tag Forward of the proposed SDR tag of types (a), (b) and (c) (explained in Fig. 1).

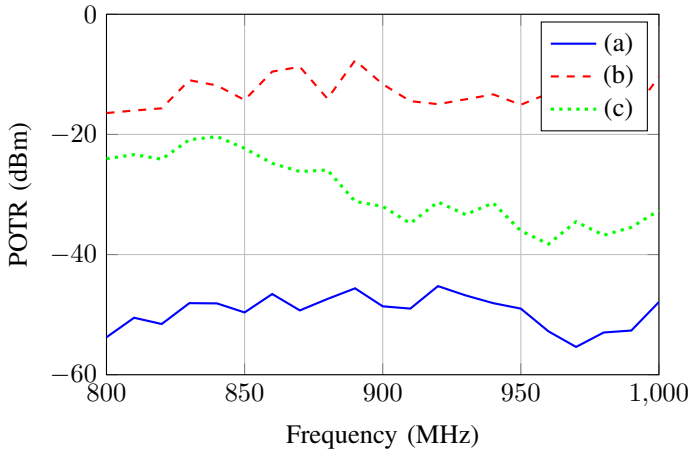

Fig. 8. Power On Tag Reverse of the proposed SDR tag of types (a), (b) and (c) (explained in Fig. 1).

Fig. 7 presents the POTF of the proposed tag for the architectures (a), (b) and (c) (see Fig.1). We can see that the POTF of the tag type (b) can be as low as -10 dBm in the considered bandwidth. By reversing modulator and envelope detector [type (a)], sensitivity can be improved to reach -20 dBm and is close to the POTF of the bistatic configuration [type (c)]. Note that this sensitivity can still be increased by using a better matching network and/or using a multi-stage rectifier to increase the voltage at the input of the data slicer.

The POTR of the proposed tag for architecture (a), (b) and (c) is presented in Fig. 8. POTR is higher than -20 dBm for type (b). On the other side, type (a) tag offer a lower POTR since a fewer power reach the modulator. Thus, the modular architecture of the tag can be used to optimize the POTR or the POTF depending on the requirements.

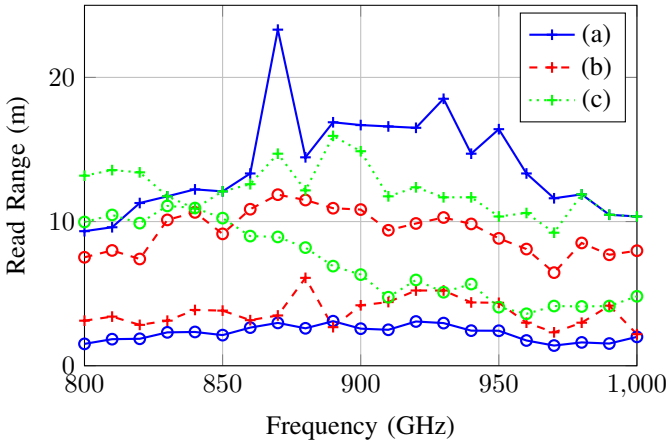

Fig. 9. Forward read range (plus markers) and reverse read range (circle markers) of the proposed SDR tags for EIRP = 36 dBm, $G_r = 6$ dBi and $P_{r\,min} = -75$ dBm.

B. Read Range in Free Space

Read range of RFID system in free space can be limited by the tag or reader sensitivity. When the RFID system is limited by the chip sensitivity, the maximum (forward) distance d_t at which a tag can be activated is given by the Friis equation. When the RFID system is limited by the reader sensitivity, the maximum (reverse) distance d_r at which a tag can be activated is given by a modified form of the radar equation [21]. Both read range in monostatic can be expressed as:

$$d_f = \frac{\lambda}{4\pi}\sqrt{\frac{P_r G_r}{\text{POTF}}} \quad d_r \leq \sqrt[4]{\frac{P_r G_r^2 \lambda^2 \sigma_d}{(4\pi)^3 P_{r\,min}}} \tag{2}$$

where σ_d its delta RCS [22]. Finally, read range of a RFID system in free space corresponds to the minimum between the forward read range and the reverse read range $d = \min(d_f, d_r)$.

Fig. 9 presents both the forward range and the reverse range of the proposed tag assuming an effective isotropic radiated power of 36 dBm (with $G_r = 6$ dBi) and a reader sensitivity of -75 dBm. Note that for this reader configuration, type (a) tag are limited by the reverse read range whereas type (b) tag is limited by the forward read range. The modular architecture of the proposed tag can allow one to optimize the performance according to the reader characteristics.

C. Sensing Capability

This tag can also offer functionality which can not be found in any classical tag. For example, the ATmega328p integrates a 12 bit Analog to Digital Converter (ADC) which can be used to measure a voltage between 0 and 5 V. The ADC can also be used measure any external sensors. For example, by connecting directly the output of the envelope detector to the ADC, the tag can estimate its received power. This information can be used in link budget analysis to estimate the transmitted power (for a fixed distance) or the distance (for a fixed transmitted power) with the reader. Moreover the ADC value can be processed and used by the tag during the inventory round. For example, this tag can dynamically adjust its sensitivity by generating the RN16 only if the value of the ADC is higher than a given software threshold. For external sensors, the value of the ADC can also be backscattered to the reader directly in the EPC field.

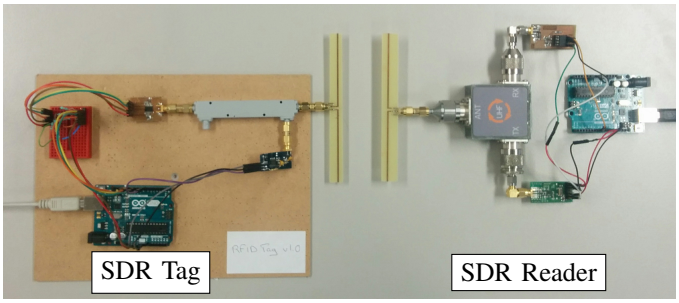

Fig. 10. Software defined tag and software defined reader [1].

D. Reading with a SDR reader

This section illustrates the possibilities of this tag when read by the reader presented in [1]. The reader has a transmitted power of $P_r = 10$ dBm (with an antenna gain of $G_r = 2$ dBi. This reader allows one to generate any command. Combined with the proposed tag, this RFID platform represents great and unique tool to study, understand and evaluate the RFID technology.

Fig. 11 presents the inventory realized by the SDR reader and the SDR tag. We can see the full inventory round which include a query command, the RN16 reply, the ACK command, and the EPC reply. RN16 and EPC have been set to all ones in software. Note that every timing information can be estimated and controlled for both reader commands and tag replies.

IV. Tag Characteristics and Future Work

As said previously, the presented tag allows one to generate any reply defined by the EPC Gen2 protocol and is able to decode any reader command in real time. Thus, this tag can theoretically take part in any inventory. Moreover, since a full access to the timing information is provided to the user, a significant quantity of information is now available.

During the inventory, current drawn is equal to 25.8 mA for the Arduino, 0.15 mA for the receiver and is negligible for the backscattering modulator. This tag can also be easily transformed into a portable tag using a battery shield while keeping an important autonomy. For example, using a 9 V battery (500 mA/h), the proposed tag can operate during 19 h.

This tag can be used also to monitor/spy any inventory round used by industrial readers. For example, this tag has been used to decode the query command send by the Tagformance Pro reader. The preamble uses a `RTcal` equal to 62.5 µs (with `L0`: 25 µs and `L1`: 43.75 µs) and a `TRcal` of 200 µs. The query command sets the divide ratio to `DR`: 8, uses the FM0 modulation (`M`: 0), with pilot tone (`RText`:

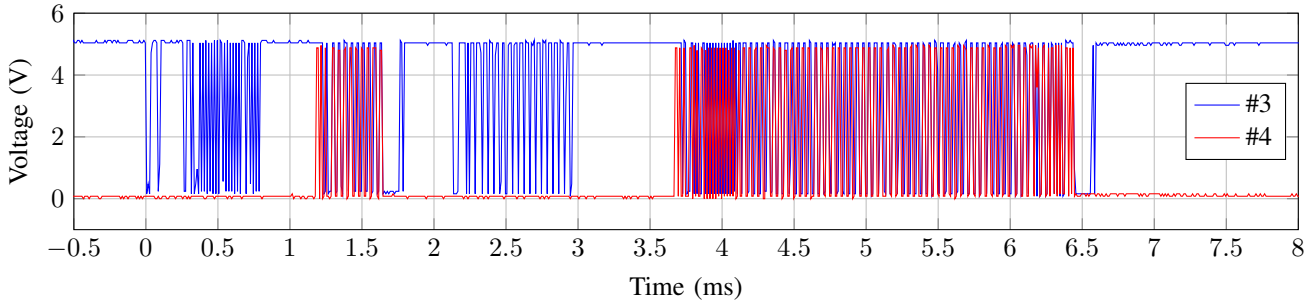

Fig. 11. Single slot inventory showing the Query command, the RN16 reply, the ACK command and finally the full EPC of the tag. RN16 and EPC are set in software to all ones. #3 is at the output of the data slicer, #4 is at switch input (see Fig. 2).

1), involving all tags (`Sel`: 0), in S0 (`Session`: 0), in state A (`target`: 0) with a Q-value of 0. These information can easily be accessed by this SDR tag. This tag can also be used to estimate complex slot count selection algorithms [3, Annex D] used by industrial readers to estimate and update the estimated tag population.

Read range using the Tagformance Pro reader can reach several meters in indoor environment. With the reader presented in [1] (with $P_t = 10$ dBm), read range in monostatic mode (using the directional coupler) is lower than 5 cm and can achieve 10 cm in bistatic. Optimization of the receiver front-end (*e.g.*, better diodes, biased detector, multi-stages) can significantly improve the read range value.

Finally, the proposed architecture can allow to design new functionalities on top of the EPC Gen2 protocol which can provide new services to the user. File transfer, or custom gateways between EPC Gen2 and different protocols (BLE, Zigbee...) can be addressed, for example. Note that both reader and tag can also be used to transmit information between each other without using the RFID standard by defining custom protocols.

V. Conclusion

In this paper, a UHF RFID tag defined in software is presented allowing to read a UHF tag in real time. The modular architecture allow anyone to reconfigure the tag and all tasks specific to the RFID protocol are entirely handled in software by the Arduino Uno. All in all, we believe that the simple open source tag presented here can become an extremely valuable research tool as well as an educational platform especially when used with in conjuction with SDR readers. We hope that these tools can bring many talented researchers to the field of UHF RFID.

References

[1] N. Barbot, R. de Amorim, and P. Nikitin, "Simple low cost open source UHF RFID reader," *IEEE RFID J.*, vol. 7, pp. 20–26, 2023.

[2] H. Stockman, "Communication by means of reflected power," *Proceedings of the IRE*, vol. 36, no. 10, pp. 1196–1204, Oct. 1948.

[3] EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Standard, Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz, Version 2.1, 2018.

[4] Beontag E62. [Online]. Available: www.atlasrfidstore.com/beontag-e62-rfid-paper-tag-monza-r6-p/

[5] R. Redemske and R. Fletcher, "Design of UHF RFID emulators with applications to RFID testing and data transport," in *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*, 2005, pp. 193–198.

[6] A. P. Sample, D. J. Yeager, P. S. Powledge, A. V. Mamishev, and J. R. Smith, "Design of an RFID-based battery-free programmable sensing platform," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 11, pp. 2608–2615, Jun. 2008.

[7] S. J. Thomas, "RFID for everyone: Design of an easily-accessible, experimental UHF RFID platform," in *2019 IEEE International Conference on RFID (RFID)*, Phoenix, AZ, USA, Apr. 2019, pp. 1–6.

[8] D. Fabbri, E. Berthet-Bondet, D. Masotti, A. Costanzo, D. Dardari, and A. Romani, "Long range battery-less UHF-RFID platform for sensor applications," in *IEEE RFID-TA Conference*, Pisa, Italy, Sep. 2019, pp. 80–85.

[9] F. Muralter, L. Arjona, H. Landaluce, and A. Perallos, "A fully customizable RFID research platform with exchangeable modules," *IEEE Sensors J.*, vol. 21, no. 13, pp. 15 379–15 385, Jul. 2021.

[10] J. Mitsugi and O. Tokumasu, "A practical method for UHF RFID interrogation area measurement using battery assisted passive tag," *IEICE Transactions on Communications*, vol. 91, no. 4, pp. 1047–1054, 2008.

[11] O. Abdelmalek, D. Hély, and V. Beroulle, "Epc class 1 gen 2 uhf rfid tag emulator for robustness evaluation and improvement," in *2013 8th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Abu Dhabi, United Arab Emirates, Mar. 2013, pp. 20–24.

[12] D. De Donno, L. Catarinucci, and L. Tarricone, "RAMSES: RFID augmented module for smart environmental sensing," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 7, pp. 1701–1708, Jul. 2014.

[13] R. Colella, L. Tarricone, and L. Catarinucci, "Spartacus: Self-powered augmented RFID tag for autonomous computing and ubiquitous sensing," *IEEE Trans. Antennas Propag.*, vol. 63, no. 5, pp. 2272–2281, May 2015.

[14] M. Buettner and D. Wetherall, "A "gen 2" rfid monitor based on the usrp," *Comput. Commun. Rev.*, vol. 40, pp. 41–47, 2010.

[15] SDR UHF RFID tag. [Online]. Available: https://github.com/nicolas-barbot/SDR_UHF_RFID_tag

[16] P. Nikitin, S. Ramamurthy, and R. Martinez, "Simple low cost UHF RFID reader," in *2013 IEEE International Conference on RFID (RFID)*, Orlando, FL, USA, Apr. 2013, pp. 1–2.

[17] S. J. Thomas and M. S. Reynolds, "A 96 Mbit/sec, 15.5 pJ/bit 16-QAM modulator for UHF backscatter communication," in *2012 IEEE International Conference on RFID (RFID)*, Orlando, FL, Apr. 2012, pp. 185–190.

[18] Arduino Uno. [Online]. Available: docs.arduino.cc/hardware/uno-rev3

[19] ADG904. [Online]. Available: www.analog.com/en/products/adg904.html

[20] ARC Program. [Online]. Available: https://rfid.auburn.edu/arc/

[21] N. Barbot, O. Rance, and E. Perret, "Classical RFID vs. chipless RFID read range: Is linearity a friend or a foe?" *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 9, pp. 4199–4208, Sep. 2021.

[22] N. Barbot, O. Rance, and E. Perret, "Differential RCS of modulated tag," *IEEE Trans. Antennas Propag.*, vol. 69, no. 9, pp. 6128–6133, Sep. 2021.