PIC-1

Automatic Inventory Checking

Project Specifications - v. 2.0

Autores:

Francisco Guilherme Plantier Rosa (102546)
francisco.g.plantier.rosa@tecnico.ulisboa.pt
Filipe Miguel Fernandes Mendes (102978)
filipe.f.mendes@tecnico.ulisboa.pt
André Filipe Henriques Marrazes (103520)
andre.marrazes@tecnico.ulisboa.pt
Duarte Daniel Mestre de Pardal(102982)
duarte.pardal@tecnico.ulisboa.pt
Miguel Tiago Elias Oliveira (103551)
miguel.t.elias.de.oliveira@tecnico.ulisboa.pt
António Pedro Patrício Fontainhas Batista de Oliveira(103515)
antonio.pedro.oliveira@tecnico.ulisboa.pt

2023/2024 – 2º Semester Group 36

1 General characteristics

1.1 Key Features

- RFID tag reader with BLE (Bluetooth Low Emission) connection to an external device
- UI via WebApp
- WebApp-reader connection via BLE through host
- 5V Supply Voltage
- Power Consumption under 10W (peak)
- Maximum range of 0.75m
- ISO18000-6C (EPC Gen2) protocol

1.2 Bill of Material

Material	Quantity	Description
AS3992 UHF RFID Reader	1	RFID reader with an associated microcontroller, providing protocol support for ISO18000-6C
RF SOLUTIONS ANT-PCB4242-FL	1	UHF antenna with 50Ω impedance matching and MMCX connector
Barrel connector to female wire connectors	1	
The Kennedy Group - SmartTherm RFID Tags	6	UHF RFID EPC Gen 2 transponder tags
Raspberry Pi 5	1	Single-board computer with wireless LAN and Bluetooth connectivity
Raspberry Pi 5 power supply	1	Power Supply, 15.3W maximum output power, USB-C connector
Female-female jumper connector	5	

1.3 System Description

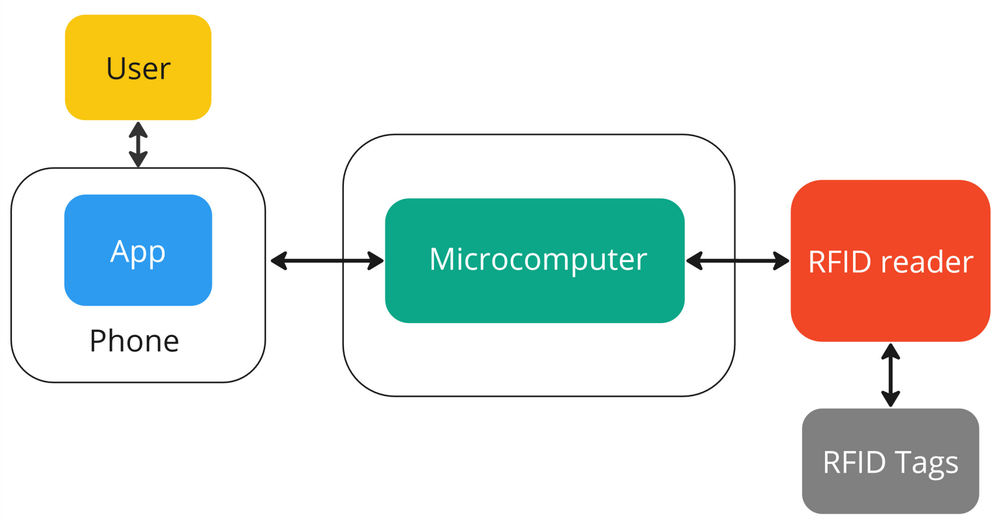

Figure 1: System Description block diagram

This system's goal is to allow easier inventory checking in closed-spaces using RFID tag detection. Using the Raspberry Pi 5 as a host for the system, two data connections are established, one between the host and the AS3992 RFID Reader (via UART, with power been supplied via USB), and the second via BLE (Bluetooth Low Emission) with a smartphone, or any other device able to connect via BLE.

After a succesful BLE connection between the host and the phone, the user selects (through the WebApp) a list of items already logged into the server, and associate to an ID and to the serial number (EPC Gen 2) of the a certain tag. That list is sent to the host, who checks the input sent by the RFID reader. Then, a list of all detected tags is sent to the host, who will compare it with the list sent by the phone. The host will mark which items are detected and which are not, send that information to the phone, relaying that information to the WebApp, informing the user of any changes.

It is required that the system is connected at all times to a power supply, able to supply 5V, at peak 1A (5W).

2 Electrical characteristics

2.1 Electrical connections

Component	Connection
AS3992 RFID Reader	Barrel connector → 5V (pin 2) e GND (pin 6), Raspberry Pi
	Tx → GPIO 15 (Rx$_d$0 - pin 10), Raspberry Pi
	Rx → GPIO 14 (Tx$_d$0 - pin 8), Raspberry Pi
	GND → GND (pin 25), Raspberry Pi
	MMCX antenna connector → RF SOLUTIONS ANT-PCB4242-FL
Raspberry Pi 5	5V (pin 2) and GND (pin 6) → Barrel connector, AS3992
	GPIO 15 (pin 10) → Tx, AS3992
	GPIO 14 (pin 8) → Rx, AS3992
	USB-C → Power Supply (230V @ 50Hz)

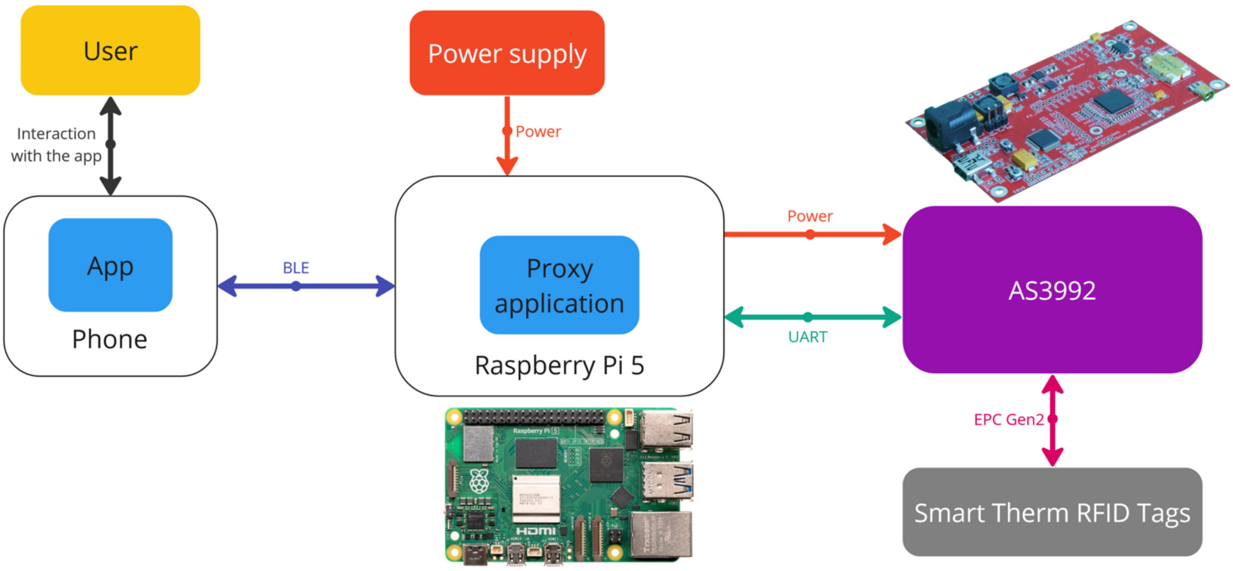

Figure 2: Diagram with the system's interactions protocols

2.2 Electrical specifications

Supply Voltage (V_s)	5V (maximum 6V)
Working power (peak) (P_{peak})	8.4W
Working power (standby) ($P_{standby}$)	4.2W
Reading Distance (r_{max})	0.75m
Tag Protocol	ISO18000-6C (EPC Gen. 2)
Tag Area (A_{tag})	90mm x 1mm
Working Frequency (f)	920MHz
RF power ($P_{t_{antenna}}$)	20dBm (0.1W) max.
RFID reader receive sensitivity ($P_{r_{minimum}}$)	-86dBm ($2.51 \times 10^{-12} W$)
Minimum power supply requirements	5V @ 3A (15W)
Antenna Gain ($G_{antenna}$)	5dBi @ 915MHz
Antenna Area ($A_{antenna}$)	49.5mm x 49.5mm

3 Maximum Values Estimates

3.1 Maximum reading distance

The maximum reading distance estimate can be found using the radar equation (or Friis formula)[2], giving:

$$P_r = A_{e_{Reader}} \cdot P_{back} \tag{1}$$

P_{back} and $A_{e_{Reader}}$ can be calculated using:

$$P_{back} = \frac{P_t \cdot G_{Reader} \cdot \sigma}{(4\pi r^2)^2} \tag{2}$$

$$A_{e_{Reader}} = \sigma \tag{3}$$

Where σ is the radar cross-section. There is a lack of information regarding the tags used. As such, a very rough estimate for the radar cross section was assumed to be $\sigma = \frac{4\pi A_{tag}}{\lambda^2}(\Gamma_{tag}^2)$[1], where Γ_{tag} is the reflection coefficient and A_{tag} is the effective aperture area. An estimate of A_{tag} can be done by taking into account the directivity D of a similar tag (Avery Dennison AD-160u7 RFID tags). Using the information in the datasheet regarding the orientation sensitivity, it suggests the presence of a half-wave dipole antenna, thus, we can assume $G_{reader} = 2.15dBi(1.64)$. Regarding $P_{t_{antenna}}$, the maximum output provided by the RFID module, is $0.1W$. Still, both the efficiency ($\eta = 55.7\%$) and antenna gain ($G_{antenna} = -2.43dB$) have to be taken in to account, so $P_1 = \eta P_{t_{antenna}} G_{antenna} = 0.047W$. As for, $P_{r_{min}} = 2.51 \times 10^{-12} W$. So the final formula, as adapted from [2] is:

$$r_{\max} = \frac{\lambda}{4\pi}\left[\frac{P_1 \cdot G_{reader}^2 \cdot G_{tag}^2}{P_{r_{\min}}}\right]^{\frac{1}{4}} = 2.86m \tag{4}$$

It should be noted that this estimate is extremely flawed, and it was just used as an approximation in the sizing. It ignores important parameters such as reflections, losses, and other effects. Besides, the presence of noise, clutter and coupling will also influence[1].

3.2 Peak and standby power consumption

The total power consumptions can be calculated by adding the individual power consumption of both the AS3992 and the Raspberry Pi, as defined in the datasheets and databases and benchmarks of usage[1]. As such, the peak power consumption is:

$$P_{peak} = P_{max_{AS3992}} + P_{max_{Raspberry}} = 2 + 6.4 = 8.4W \tag{5}$$

And the standby power consumption:

$$P_{standby} = P_{standby_{AS3992}} + P_{standby_{Raspberry}} = 1.7 + 2.5 = 4.2W \tag{6}$$

4 Communication between devices

The Raspberry Pi acts as a proxy between the app running on the user device and the AS3992 RFID module, forwarding requests and responses and doing some basic processing.

4.1 Between the app and the Raspberry Pi

The protocol connecting the application with the Raspberry Pi is built on top of Bluetooth Low Energy's Generic Attribute Profile (GATT) protocol. The Raspberry Pi acts as the GATT server, waiting for user devices to connect to it. To achieve bidirectional communication, the user device sends a message by writing to a GATT characteristic and the Raspberry Pi sends a message by notifying the user device of an update to a GATT characteristic.

The protocol defines the tag list request and response messages, the closest tag request and response messages, along with the error message. A formal specification of this protocol is available here.

4.2 Between the Raspberry Pi and the RFID module

The Raspberry Pi communicates with the RFID module via UART using RFID module's protocol. A summary of the relevant messages for this project can be consulted here.

5 Operations

5.1 Tag list and closest tag requests

Whenever the user requests an inventory, the app requests the tag list from the Raspberry Pi and whenever the user tries to add a tag to the database, the app requests the closest tag from the Raspberry Pi. When the Raspberry Pi receives one of these requests, it requests the RFID module perform a read of the tags in range and forwards the returned tag list to the user device. To improve reliability, multiple read requests may be issued, in which case the Raspberry Pi sends a tag list containing the ID of all tags present in at least one of the lists.

[1]As mentioned in https://www.pidramble.com/wiki/benchmarks/power-consumption

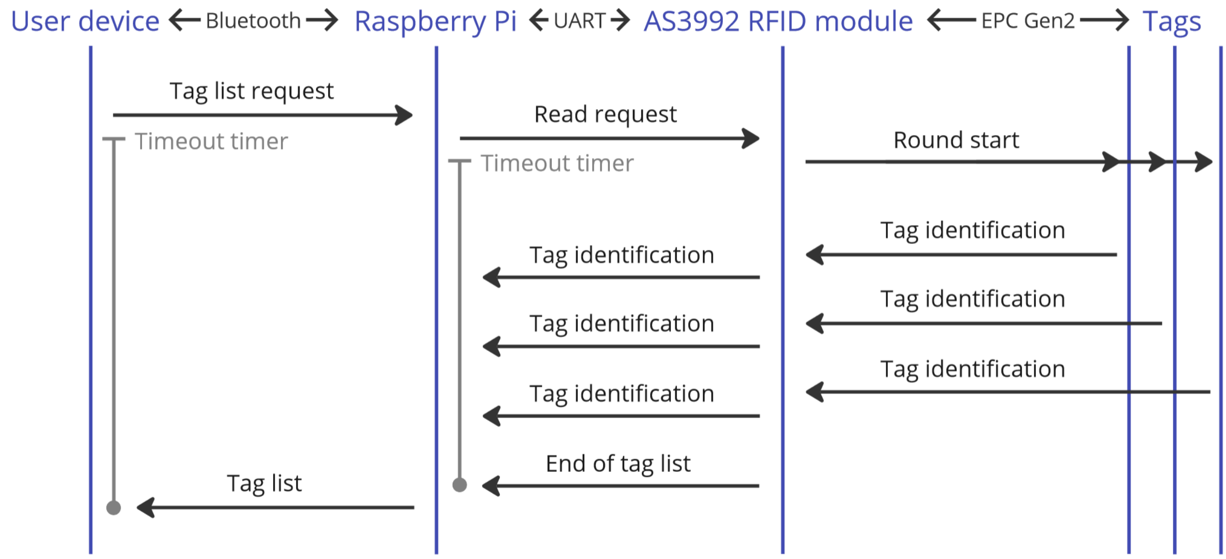

Figure 3: Communication for a tag list request

For closest tag requests, only the tag with the highest RSSI is sent to the user device.

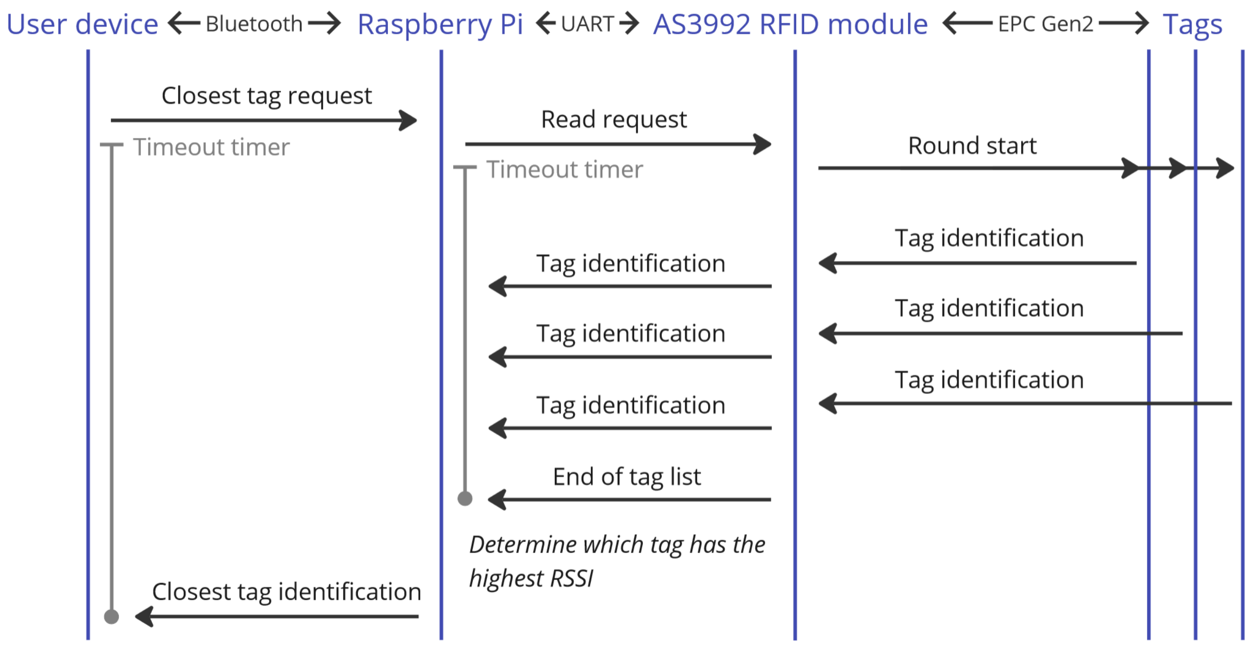

Figure 4: Communication for a closest tag request

If it is taking too long to receive a reply from the RFID module or if an invalid message is received, the Raspberry Pi sends an error to the user device. The most common cause of these errors are a bad physical connection between the Raspberry Pi and the RFID module.

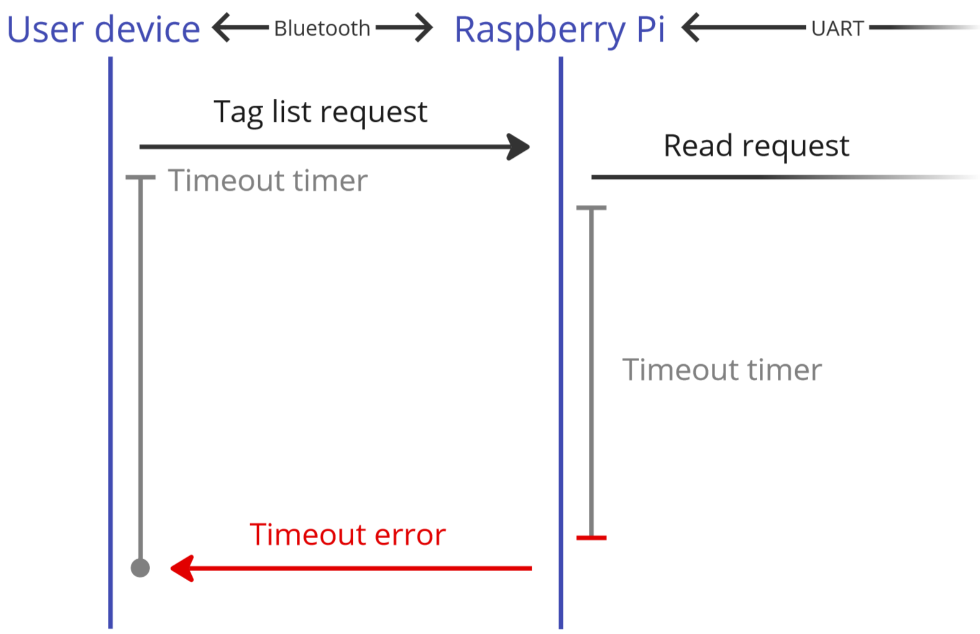

Figure 5: Communication in case of a timeout error

The RFID module may also respond to a read request with an error, in which case a RFID error message is sent to the user device.

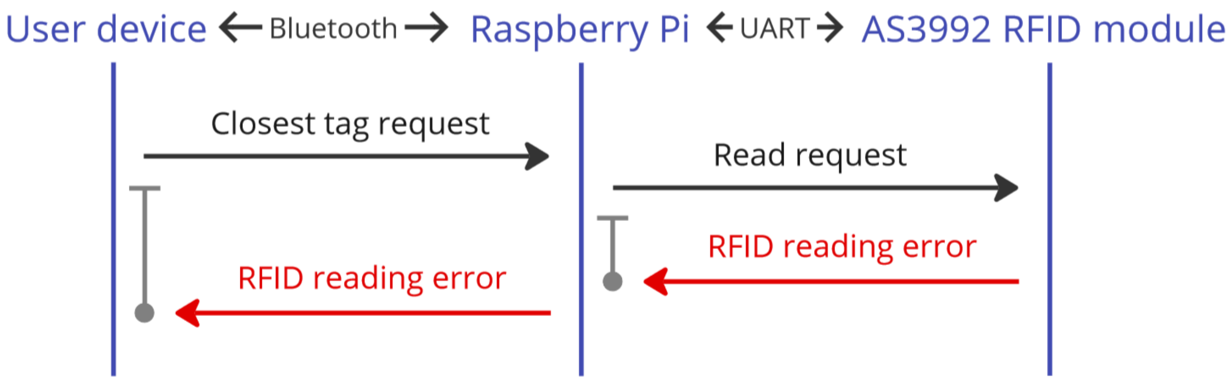

Figure 6: Communication in case of an RFID error

5.2 Initialization

When the Raspberry Pi boots up, it configures the RFID module and turns on the power to the antenna.

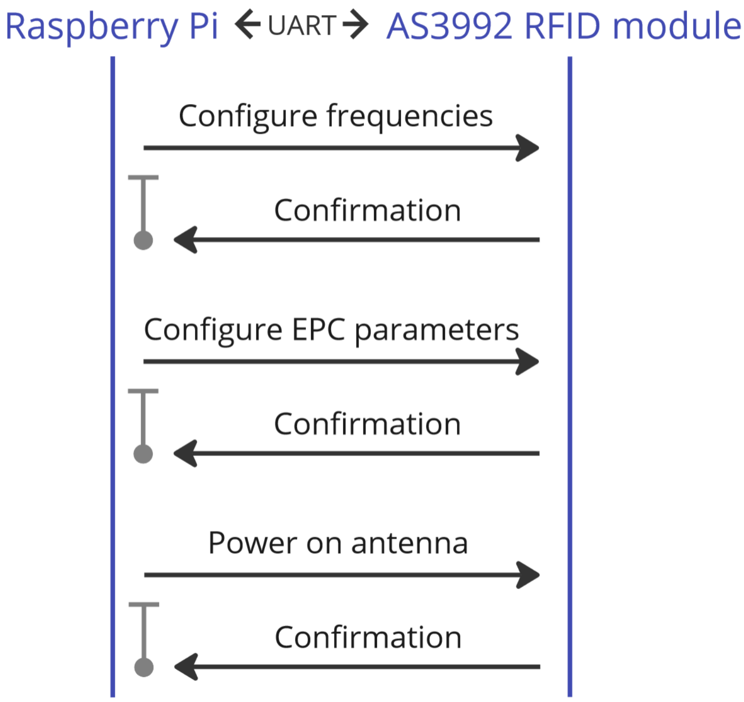

Figure 7: Communication for when the Raspberry Pi boots up

6 Results

To test the performance of the system, for each distance between 0 and 90cm in increments of 5cm, 40 measures were taken repeatedly, in the same conditions. Originally the average of the RSSI (returned signal strength indicator) values was used as a metric of the proximity of a tag to the antenna. However, the unit of this parameter was not provided by the documentation of the reader, being represented by just 4 bits for the in-phase and quadrature components, each. This leads to a possible range of 0 through 15 for each component with no known absolute scale. Still, as expected, both components exhibit a downward trend as the distance increases.

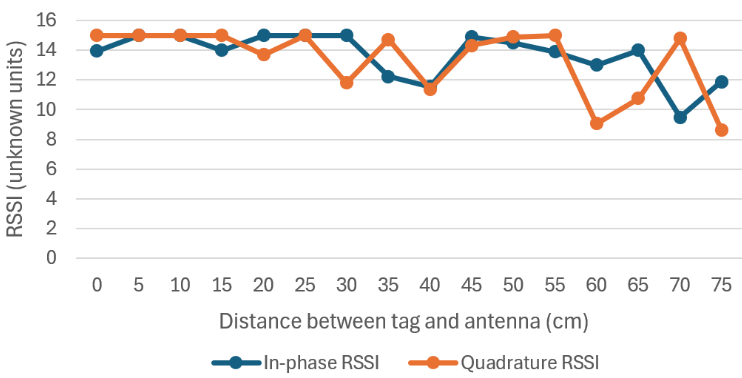

Figure 8: Average RSSI results for air tests

Instead, the success rate of the detection was used as the metric, registering the percentage of success of the 40 measures performed. Besides testing for open range (air), other materials were used to test the performance for various thickness, such as wood, glass, styrofoam, low density polyethylene plastic, cardboard and metal (aluminum), with different success rates.

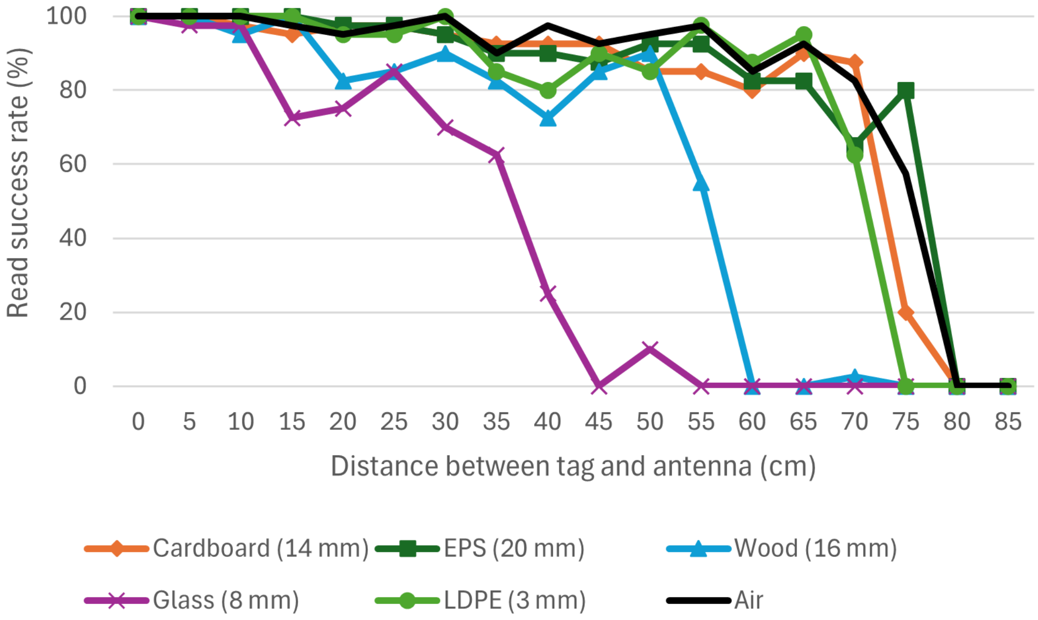

Figure 9: Experimental read success rate (probability of a single successful read) for various materials

By default, our system performs 20 RFID reads for each user-initiated inventory (tag list) request. As such, admitting that the probability of a single read being successful (r) is independent of all other reads — an empirically verified premise —, we can establish the following relationship between r and the probability of at least one of the 20 reads being successful (p), which translates into a successful tag detection:

$$p = 1 - (1 - r)^{20}$$
$$r = 1 - \sqrt[20]{1 - p}$$

To ensure a successful detection with at least 99% probability, the probability of a single successful read must be at least 20.6%.

References

[1] Michele Borgese, Simone Genovesi, Giuliano Manara, and Filippo Costa. Radar cross section of chipless rfid tags and ber performance. *IEEE Transactions on Antennas and Propagation*, 69(5):2877–2886, 2021.

[2] Klaus Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communications*. Wiley and Sons, 3rd edition, 2010.