4B25 CW#5 - Design of a Low Cost RFID Sensor

Ryan Jones | Sidney Sussex College | rlj36@cam.ac.uk

Project Description

RFID (Radio-frequency identification) is a technology which utilises electromagnetic fields to identify and track tags attached to objects. This is an incredibly powerful tool, being used frequently in retail stores, secure access to buildings, and many more applications [1]. The aim of my project was to develop a low-cost embedded system designed for recognising and saving an RFID tag, and being able to verify that this tag is currently saved and identifiable by the microcontroller.

The problem solved here is designed to be used in a security/access control building environment, where an administration staff can use this tool to add a new tag to the system and use an additional function to verify that the tag is correctly added and recognised. This, however, is a particularly interesting solution compared to existing systems as it uses a microcontroller which can interface with a variety of other sensors. Given more time/budget, an engineer installing this system into a building could interface this with existing infrastructure such as a facial/fingerprint detector and this highlights the interface-ability of the system which (along with low-cost) gives it an inherent advantage.

Current State of the Art

RFID technology has developed greatly recently, but still suffers a crucial limitation in its high cost implementation for most companies [2]. The technology is particularly ubiquitous in the access control market, but involves more expensive readers with very complex read/write protocols to the tags in order to ensure there is little card tampering/cloning. The aim of this project is not to mirror this state of the art technology with its complex control and security protocols, but instead develop a low cost solution available to many industries for interfacing with existing systems. The RFID reader itself is very cheap already, but by using a microcontroller then it opens up a very broad spectrum for which this can be extended to by involving other sensors. Figure 1 shows a new state of the art chipless RFID tag printed with inkjet conductive ink. This is the potential future of the industry, as the design and fabrication of the integrated circuit are large factors of the overall sensor cost. Therefore, it will give drastic improvements to costs when the technology is fully developed.

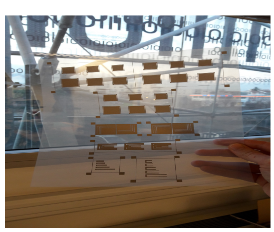

Fig. 1. Chipless RFID tag - Source: Wikipedia Chipless RFID

Project Approach

For my project, I have used the given FRDM-KL03 Evaluation Board alongside a purchased MFRC522 RFID sensor breakout board (as shown in Figure 2), which is a highly integrated reader/writer IC for contactless communication at 13.56 MHz. It is a very cheap sensor (£5.29) and comes with accompanying key ring and card RFID tag. It operates on 3.3V, and has an SPI interface which connects to the FRDM-KL03. The driver interfaces the FRDM-KL03 board by using the Warp menu to provide option #1) saving the tag's UID (**U**nique **Id**entifier) by reading from the SPI bus to determine whether the request tag matches the reference key set on the RFID tag. And #2) scanning for a tag and seeing if the saved UID from option #1) matches the detected RFID tag.

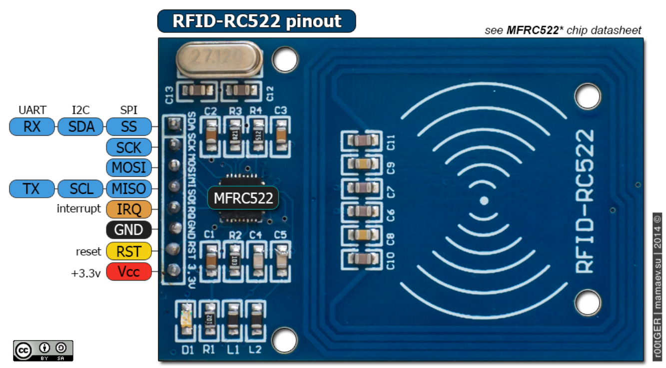

Fig. 2. MFRC522 RFID Sensor Pinout

I achieved this by creating files *MFRC522.c* and *MFRC522.h*, which will form the basis of the RFID functions to be used with the *warp-kl03-boot.c* file to perform these functions. These files were based upon existing Arduino and C libraries for the MFRC522 [4] [5] [6], which were open source Github repositories implementing the MFRC522 in either an Arduino or Atmel microcontroller. This involved the transformation of these existing libraries into the Warp-firmware to create *MFRC522.**. The *warp-kl03-boot.c* file was modified such that the '#' key would enter the "RFID Menu", which contained function #1) and #2). Function #1) scanned for the presence of a card and would save the tag's UID to current memory if present (would print "No card present" if not). #2) would again scan for a tag, but check whether the presented tag's UID matches the currently saved tag. A further extension, which I would have liked to have added, would be the addition of a text file (I believe it would require quite extensive FRDM-KL03 JTAG knowledge to do this) which the program would then save these UIDs to. This database could then be crossreferenced in #2), rather than just storing a singular UID at a time. In a real world system, this could be achieved using an external storage but the KL-03 doesn't appear to have any internal storage available.

Figure 3 shows my practical implementation, including the INA219 current sensing device for the result section below. Alongside this (top left) are the RFID tag and card provided with the MFRC522 RFID sensor. The code attached with this report includes: *MFRC522.**, *warp-kl03-boot.c* (changed), and the Github repository including the remaining files such that the project can be easily compiled for any user who wishes to try it. This repo can also be used to see the git difs if compared to the master files it was forked from. All outputs are printed to the console, and there is no physical indication when a card has been scanned or verified. An accompanying video is uploaded to help show the implementation in practice.

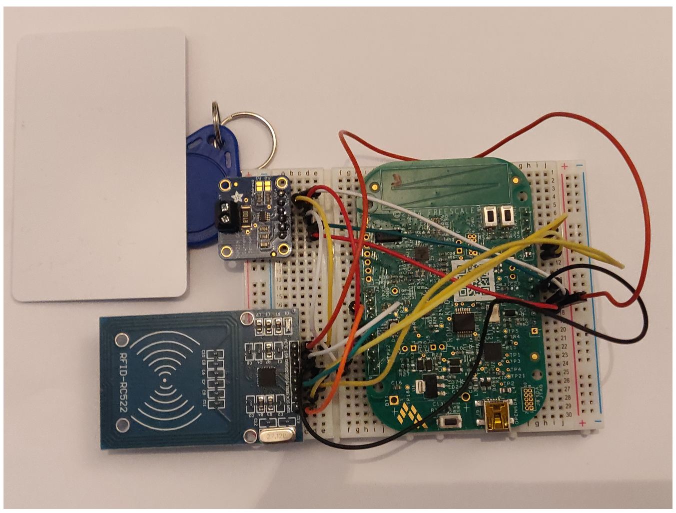

Fig. 3. Design implementation for my project - including the FRDM-KL03 evaluation board, MFRC522 RFID sensor and INA219 current sensor, and RFID card/tag

Results

The result of this project is a fully functioning RFID sensor, with the potential for many further extensions in the coding. I have used the FRDM-KL03 board, and developed a driver to interface with the MFRC522. Figures 4 & 5 both show the current consumption of the RFID sensor during idle and operation mode respectively. You can see that there is very little difference between both modes, and it falls well within the maximum current value from the data sheet (100mA) [3].

This was achieved using the code from Coursework 4, which implemented the INA219 current sensor with the FRDM-KL03 evaluation board to measure 1000 current values, which are plotted below.

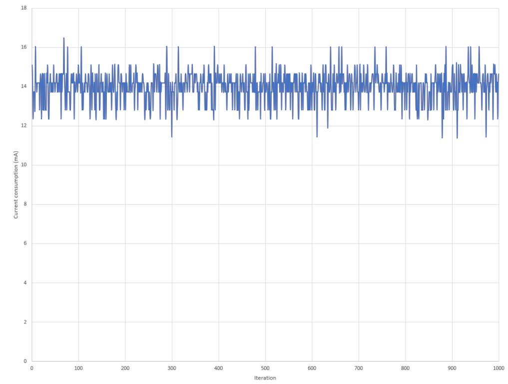

Fig. 4. Current consumption of the MFRC522 during idle mode

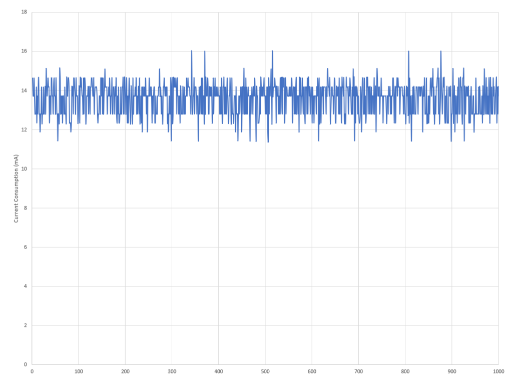

Fig. 5. Current consumption of the MFRC522 during operation mode

Conclusions

We have successfully developed a low cost (£5.49 for the MFRC522) RFID verification system using a simple FRDM-KL03 evaluation board, capable of interfacing numerous other sensors depending on the application. There are further improvements I would have like to have made, such as being able to communicate with a master database when reading (or writing from) the RFID tag - however this project was challenging in the coding aspect when porting existing libraries to the Warp firmware.

References

[1] James Thrasher, *How is RFID Used in the Real World*, https://www.atlasrfidstore.com/rfid-insider/what-is-rfid-used-for-in-applications/

[2] P. J. Soares, C. Oliveira, G. Morales, J. Arica and I. Matias, *State of the Art on Arduino and RFID*, Production Engineering Laboratory (LEPROD), State University of Northern, Rio de Janeiro

[3] NXP Semiconductors, *MFRC522 Product Data Sheet*, https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf

[4] Miguel Balboa, *Arduino library for MFRC522 and other RFID RC522 based modules*, https://github.com/miguelbalboa/rfid

[5] Bjarte Johansen, *Arduino RFID Library for MFRC522 (13.56 Mhz)*, https://github.com/ljos/MFRC522

[6] Asif Mahmud Shimon, *MIFARE RC522 module library for Atmel MCU*, https://github.com/asif-mahmud/MIFARE-RFID-with-AVR