RFID Access Control System
RFID Access Control System

Introduction
I propose to build an RFID access control system. The project will use an RFID reader to verify an RFID tag, and then send a signal to unlock a door. When the user places the tag within reading range of the antenna, the reader will process the tag ID and send it to the microcontroller. If the tag is verified to be valid, then a lock will be signaled to unlock. After a short amount of time, the lock will automatically be re-engaged. The system will be able to add and remove IDs, create a timestamp for tracking activity, and have a keyed backup in case of system failure.

Hardware Description
The RFID access control system will be built around a Cypress CY8C29466 PSoC. When making the decision about which MCU to use, I was most interested in GPIO, core size, FLASH size, RAM size, and on-chip interfaces. Several options were available from different manufacturers including the 9S08 from Freescale, the ATTINY88 from Atmel, the PIC18F2331 from Microchip Technology, and the CY8C29466 from Cypress. While all of these MCUs meet, or exceed, my project needs of 24 GPIO, 8-bit core, at least 32KB of FLASH, at least 2KB of RAM and I²C, I chose the PSoC from Cypress. Since I have previously worked with Freescale hardware, I wanted to try something different. The extensive online technical references and user friendly IDE of Cypress will allow me to experiment with and learn the new hardware without hindering the development of the project.

The other major component of the RFID access control system is the RFID reader. For this component, I’m going to use an OEM chip and antenna from SONMicro. The SM125 module is designed for 125 KHz RFID applications. It performs all of the RFID processes such as amplifying, filtering, demodulating, decoding, and digitizing. The data is then ready to be sent over I²C to the PSoC for evaluation.

The rest of the RFID access control system includes an LCD, keypad, serial connection, LED, buzzer, electronic lock, RFID tags, and an external crystal oscillator for timing accuracy. The passive RFID tags will be powered and read by the RFID module using the low frequency 125 KHz protocol, defined by ISO 18000. The LCD, keypad, serial connection, LED, buzzer, lock and crystal will all be connected to the PSoC using GPIO. With the LCD running in 4-bit mode and a GPIO used to signal a successful tag read, the total system IO used would be 22 of 24. Visual representations of the project may be seen in figures 1 and 2.

The memory requirements for the system software are relatively small. The 32KB of FLASH is more than enough to hold the program that manages the system. No more than 5KB of FLASH should be needed for the system software. The activity log has the possibility of being very large. The remaining FLASH should be more than sufficient for the system to handle about 3500 reads before the data would be written over. This is equivalent to 250 reads per day for two weeks, assuming each log entry is 6 bytes or less. A ring buffer will be used to store the ID and timestamp data. This ensures that the oldest data is the first to be written over. Retrieving the data from the system, using a serial connection to an external device, every two weeks, would likely prevent any data loss. Realistically, the data would only need to be retrieved if there was an incident that required a log of the system usage.
The final, and most critical, component of the system is the power supply. This consists of an AC adapter, voltage regulator (VR), switch mode pump (SMP), and low voltage detector (LVD). The VR would be capable of receiving 3 to 18 volts at the input while delivering 500 milliamps at 5 volts to the system. The LVD and SMP are built into the PSoC. When the input voltage drops below 5 volts, the LVD signals the SMP to turn on. This activates a boost converter to deliver a constant 5 volts to the system, regardless of the input voltage. The SWP will continue to function until the input voltage source falls below 1.2 volts.

Figure 1 Hardware Sketch
RFID Access Control System

**Figure 2** System Block Diagram

**Software Description**

The majority of the software for this project will be developed using the C programming language. A small portion will need to be written in assembly. After an initialization sequence, an endless loop will be entered. From here, different functions and interrupts will be run based on system activity. Prewritten modules will be used for the PSoC to communicate between the LCD and keypad using GPIO, as well as the RFID reader module over I²C. Additional functions will need to be written to control the LED, buzzer, and electronic lock.
User Interface

The user interface for the RFID system is fairly simple. Under normal operation, the user will only need to place their tag within reading distance of the antenna. For maintenance of the system, the administrator will press a key on the keypad to enter the maintenance state. A security code will then have to be entered on the keypad. If it is not a valid code, the system will return to the ready state. If the security code is validated, the maintenance state is entered. From here, the administrator can add users, set the time, or download the activity log. Adding users consists of entering a unique 8 bit ID number which is then stored in a look-up table. The time will be in 24 hour mode and the date will be in month, day, and year format. The activity log will be downloaded over the serial port to a terminal emulator, on a PC, when the user enters the empty data state. When the maintenance is complete, the system returns to the ready state. See figure 3 for a state diagram of the user interface.

Figure 3 User Interface State Diagram
RFID Access Control System

Communication Protocol and Standards
The RFID system will use the low frequency standard of 125 KHz, defined by ISO 18000-2, to communicate between the reader and the tag. The maximum read range of the reader is 12cm.

Development Plan
It is important that I continue to regularly work on this project throughout winter and spring quarter in order to ensure a successful end product. My main goals during winter quarter are to gather and assemble all of the hardware along with learning the PSoC IDE. Once I feel like I have a basic understanding of the development environment, I will begin to write the functions needed to control the hardware of the system. See figure 4 for a schedule of winter quarter.

The beginning of spring quarter will be a continuation of writing software to control the various hardware of the system. The whole system will then be assembled and tested. Dates must be met during spring quarter for system hardware, system software, and code reviews. The culmination of the project is the presentations during the last week of the quarter. See figure 5 for a schedule of spring quarter.

![Figure 4 Winter Schedule](image)

![Figure 5 Spring Schedule](image)
RFID Access Control System

Development Hardware and Software
I will use the CY3210-PSoCEVAL1 evaluation board with a CY8C29466 PSoC to develop the RFID access control system. The advantage of using this board is that most of the components that I need already built onto it. These include a voltage regulator and a socket for a PSoC with easy access to the ports. LEDs, breadboard, buttons, and a programming port for use with the PSoC Programmer software are also included.

PSoC Designer will be used as the IDE for this project along with a Hi-Tech brand C compiler, which is compatible with PSoC Designer. This software is available as a free download. I will be installing it on my laptop so that I can work on the project both at home and on campus.

Prototype Description
My demonstration prototype will be built around my Cypress evaluation board. A custom PCB will not be used in my demonstration because of time and resource constraints. The evaluation board contains almost all of the resources necessary to make the project fully functional. The whole project, with the exception of the electronic lock, will be housed in a plastic enclosure. The LCD, keypad, LED, buzzer, and antenna will be mounted on the surface. The evaluation board and RFID reader will be enclosed beneath the surface items. The plastic enclosure will be separate from the door; it would normally be mounted on the wall next to the door. Wires will then be run from the unit to the lock, on the door, to control the locking mechanism.

To demonstrate my RFID access control system, I plan on constructing a small door with an electronic lock attached to my project. To show how the system works I will first add an ID to the valid list. Next, I will place a valid tag within the range of the antenna to show how the system unlocks the door. An invalid tag will then be placed within the range of the antenna to show that the door only unlocks for valid tags. Then I will delete a previously valid ID and place the corresponding tag in the range of the antenna to prove that the ID does not unlock the door. In all cases, the door will open freely from the inside and automatically relock a set amount of seconds after unlocking. Finally, I will connect my project to a terminal emulation program, using the serial port and UART module, to demonstrate the downloading of the activity log.

Sustainability Design
The main aspect of the RFID system that could possibly pose a negative environmental impact is the RFID tags. Passive RFID tags can be used over and over again and can last for many years without being replaced. Modern RFID technology is more eco-friendly and meets current environmental regulations.
RFID Access Control System

Project Specifications
- RFID Communication Protocol: ISO 18000
- Serial Communication Standard: RS-232
- RFID Operating Frequency: LF 125Khz
- Crystal Operating Frequency: 32Khz
- Maximum Read Range: 12cm
- Maximum Number of IDs: 256
- Bytes Stored per Read: 6 Bytes
- Maximum Number of Log Entries: 3500 Entries

Power Requirements
- Source: AC Adapter, 3 W max output
- Worst-Case Power Dissipation: 810 mW
- Requirements Met: UL, FCC, RoHS compliant

Environmental Requirements
- Temperature Range: 0° to +85° C

PCB Size Limits
- Evaluation Board: 75mm (W) x 145mm (L) x 20mm (H)
- Enclosure: 100mm (W) x 150mm (L) x 50mm (H)

Table 1 Preliminary Parts List

<table>
<thead>
<tr>
<th>Item</th>
<th>Part#</th>
<th>Mfr.</th>
<th>Qty.</th>
<th>Price</th>
<th>Current(max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSoC</td>
<td>CY8C29466</td>
<td>Cypress</td>
<td>1</td>
<td>$8.78</td>
<td>14 mA</td>
</tr>
<tr>
<td>RFID Reader</td>
<td>SM125-M1-232</td>
<td>SONMicro</td>
<td>1</td>
<td>$10.90</td>
<td>45 mA</td>
</tr>
<tr>
<td>Antenna</td>
<td>IND125-1</td>
<td>SONMicro</td>
<td>1</td>
<td>$3.90</td>
<td>1.5 uA</td>
</tr>
<tr>
<td>Proximity Card</td>
<td>EM4100-ISO</td>
<td>SONMicro</td>
<td>3</td>
<td>$2.90</td>
<td>-</td>
</tr>
<tr>
<td>Proximity Keyfob</td>
<td>EM4100-KEY</td>
<td>SONMicro</td>
<td>3</td>
<td>$3.00</td>
<td>-</td>
</tr>
<tr>
<td>LCD</td>
<td>LCM-S01602DTR/A</td>
<td>Luminex</td>
<td>1</td>
<td>$9.28</td>
<td>5 mA</td>
</tr>
<tr>
<td>Keypad</td>
<td>96AB2-102-R</td>
<td>Grayhill</td>
<td>1</td>
<td>$12.26</td>
<td>-</td>
</tr>
<tr>
<td>Electronic Lock</td>
<td>FE595VCAM619ACC</td>
<td>Schlage</td>
<td>1</td>
<td>$128.00</td>
<td>73 mA</td>
</tr>
<tr>
<td>LED</td>
<td>-</td>
<td>Radio Shack</td>
<td>1</td>
<td>$0.25</td>
<td>20 mA</td>
</tr>
<tr>
<td>Buzzer</td>
<td>-</td>
<td>Radio Shack</td>
<td>1</td>
<td>$4.49</td>
<td>5 mA</td>
</tr>
<tr>
<td>Resistors</td>
<td>-</td>
<td>Radio Shack</td>
<td>1 Pkg</td>
<td>$6.49</td>
<td>20 mA</td>
</tr>
<tr>
<td>Crystal Oscillator</td>
<td>ECS-13</td>
<td>ECS</td>
<td>1</td>
<td>$0.89</td>
<td>-</td>
</tr>
</tbody>
</table>

*All parts except the keypad and crystal have been acquired.