Schnorr Protocol Remote Access Fob

Michael Gamota

Pedro Ocampo

Vasav Nair

Final Report for ECE 445, Senior Design: Fall 2024

TA: Pusong Li

December 11, 2024

Project/Group 35

Professor: Cunjiang Yu

Abstract

Our project is a cryptographically secure remote access system. The system consists of 2 modules: the fob and the verification and control unit (VCU). The 2 modules communicate wirelessly. The cryptographic method employed is Schnorr Identification Protocol, which is an interactive zero-knowledge proof of knowledge exchange. When the VCU has verified that the fob sending the “open” command is an authorized user, the VCU actuates a small motor, simulating a garage door. In this report we will discuss the workings of Schnorr Identification Protocol, and the hardware required for this secure remote access system. Ultimately, our system worked as intended, defined as the ability of the VCU to reliably receive and authenticate valid messages, actuate the motor in reasonable time, and resist replay attacks.

Table of Contents

Introduction

Motivation

Remote access systems are susceptible to several types of attacks based on their security topology.

Table 1. Remote access system vulnerabilities

Security Topology	Fixed Code	Passive Key	Rolling Code
Attack	Replay	Relay/Skimming	Rolling Jam

The replay attack is a vulnerability for many remote access systems, especially older garage doors. The attack involves a malicious third party using an RF receiver/emulator to intercept and record an "open" command and reusing that command to gain access later.

Passive keys, particularly those which are not encrypted, are vulnerable to skimming attacks, where a malicious third party scans the key without your knowledge and stores your data for later use. Even encrypted passive keys can be vulnerable to relay attacks where the range of a vehicle's passive key scanner is artificially extended to scan your passive key without your knowledge, unlocking your car.

Rolling code security was implemented to prevent replay attacks, however a relatively simple attack still exists. A rolling code means that every "open" command is calculated based on the previous number of valid transactions, making the next code deterministic. The attack follows: the user sends "Code 1" to the garage door or car, which a malicious third party jams and records. Since the garage or car did not open, the user sends another code, "Code 2". The malicious third party also jams this code and records it, while also sending out "Code 1", which the garage or car is still expecting as the valid code since it never received it. This causes the garage door or car to open, leaving the user satisfied and the third party with "Code 2", the next valid code, to use at a later time.

Therefore, all these remote access technologies have vulnerabilities which are easily exploitable. We find this to be unreasonable as access to one's house or car is something that should be protected with the highest level of security.

Solution

Our solution is to design an active remote access system which leverages Schnorr Identification Protocol, an interactive zero-knowledge proof of knowledge exchange, making it resilient to replay, relay/skimming, and rolling jam attacks. We will design the hardware for a key fob and verification and control unit (VCU).

The fob will have a public key and private key, which can be thought of as a known serial number, and an associated and mathematically related secret serial number. There will be an interaction between the key fob and the VCU, the fob will send a message to the lock/opener, announcing its public key and the desired command. The lock/opener will respond with a "challenge", which is a random number. The key fob will then compute the response to the "challenge" which requires knowledge of the secret key. This response is sent back to the lock/opener which can then mathematically verify if the fob has disclosed its true public key in the first message. This is done by knowing the relationship between the public and private keys. If the public key is verified and it is on the list of authorized users stored in the VCU, the verification and control unit will unlock the car/open the garage door. Something important to note is that Schnorr identification protocol is a zero-knowledge authentication scheme, which means that no information about the value of the secret key can be gained by a third party listening in on the exchange or by the VCU. The devices will communicate wirelessly at 315 MHz, a common frequency for remote access systems, particularly garage doors.

For the purposes of our demonstration, the VCU will be connected to a small motor which will spin when a valid exchange occurs.

Visual Aid

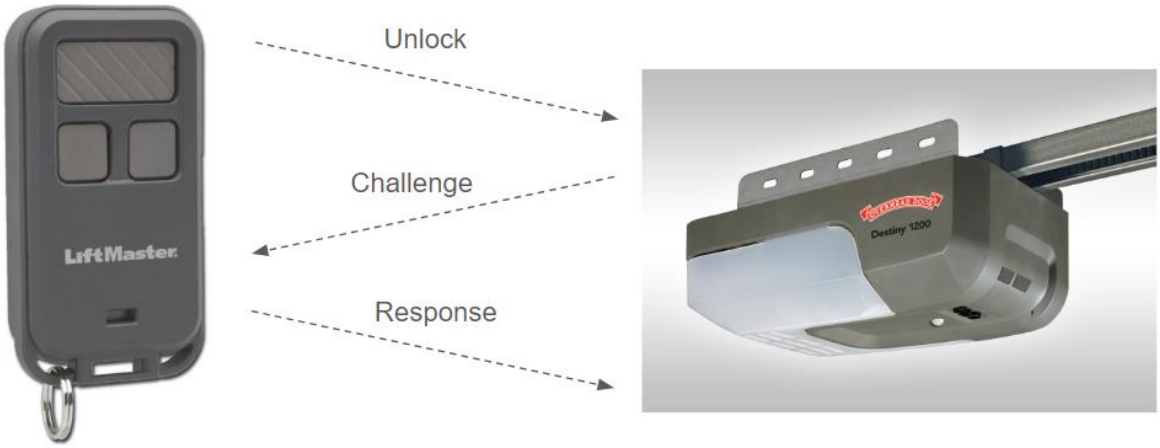


Figure 1 is an example of how our project would work at a high level.

High Level Requirements

- The signals must be detected and received from 3 meters and the signal integrity must be good enough that messages are able to be authenticated.
- The time between the unlock signal being sent and the motor spinning must be less than 2 seconds.
- A replay attack is not successful.

Design

Block Diagram

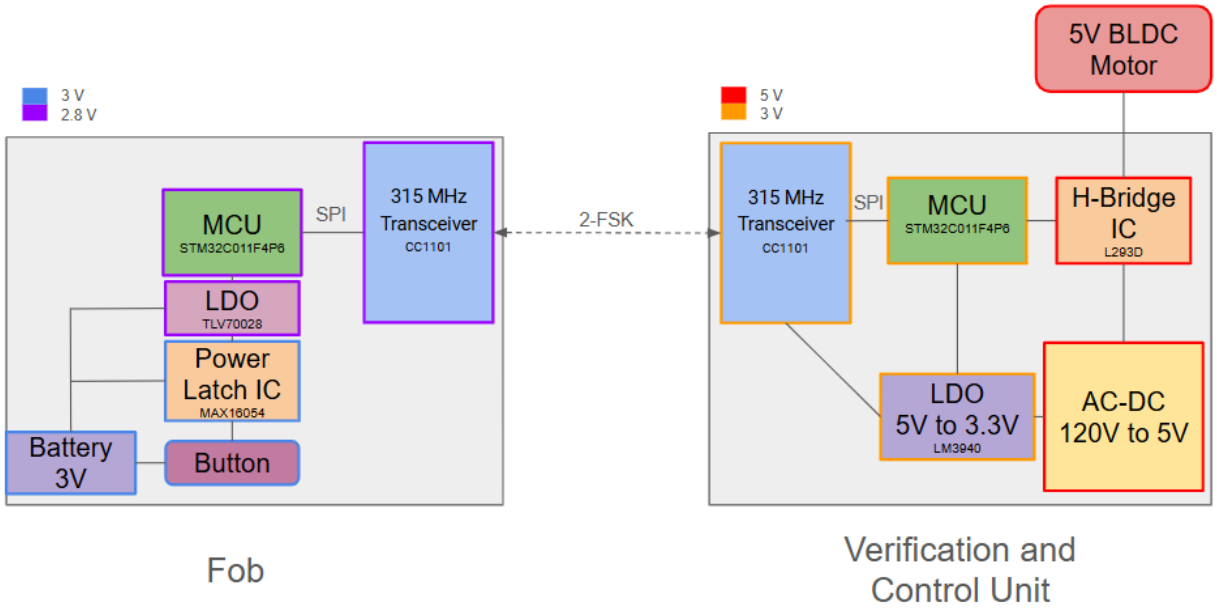


Figure 2 shows what our main components and subsystems will look like in our project, with the two main systems being the Fob and Verification & Control Unit

Procedure

The design procedure started with the definition of system level requirements. We needed to create 2 subsystems, the fob and VCU, which could communicate wirelessly and compute the required values to engage in Schnorr Identification Protocol. We also need the fob to be portable, so we decided on battery power. The VCU did not need to be portable so we opted to design an AC-DC converter so we could power the VCU via a standard wall outlet. We wanted to optimize our fob for power and size, while we wanted to optimize the VCU for signal strength.

Mechanical Design

Our fob and VCU both have 3D printed enclosures which make them more durable and portable. The fob is small enough to fit in someone's pocket easily, limited mostly by the size of the antenna and battery that we used.

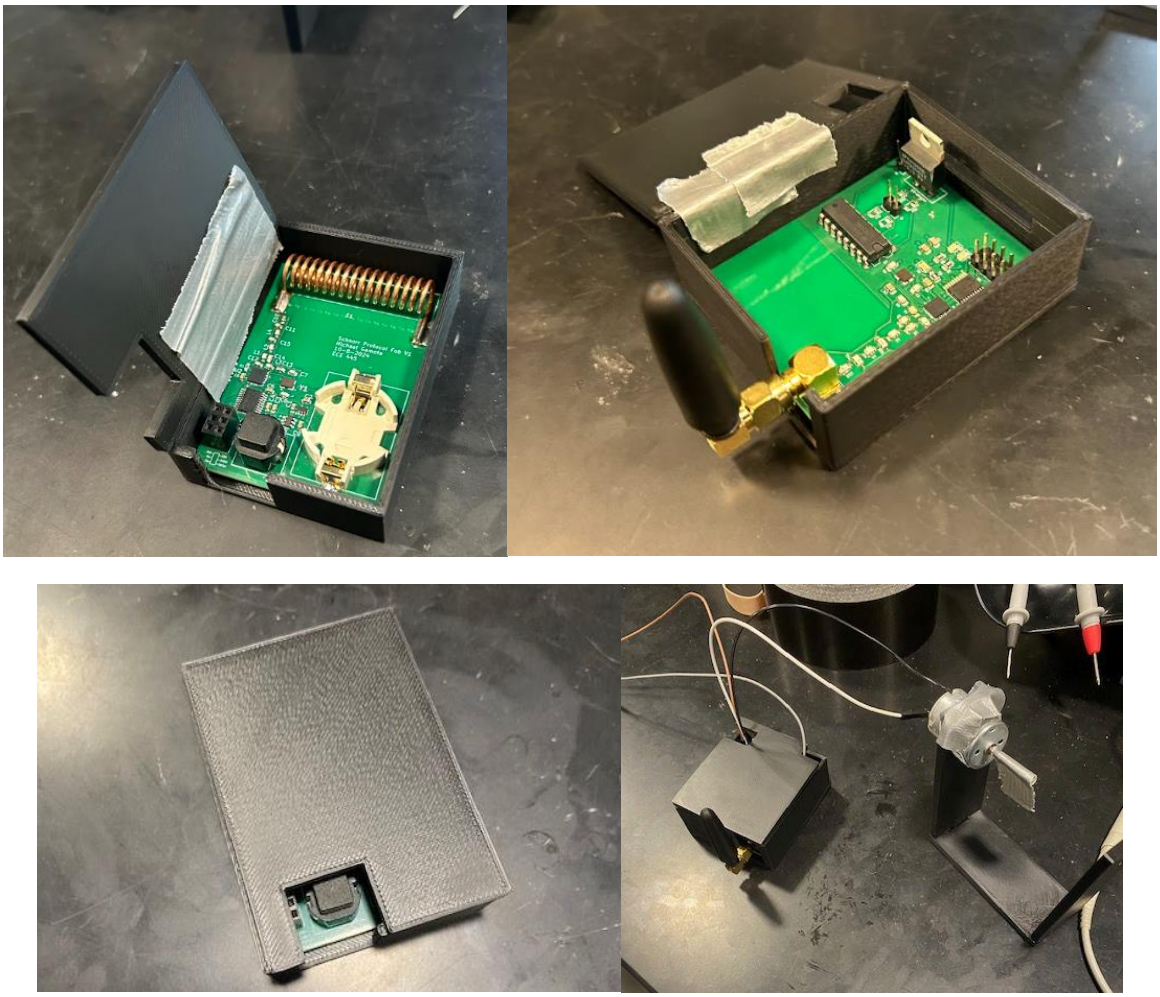


Figure 3. shows the housings with the lids opened and closed (Left: Fob, right: VCU)

Subsystem Design and Verification:

RF Transceiver

Identical RF transceiver subsystems are present on both the fob board and the VCU. The RF transceiver communicates with the microcontroller via an SPI interface, on which the RF transceiver is the only slave device. Full-duplex (simultaneous two-way communication possible) SPI requires 4 connections between the master, the STM32 MCU [4], and the slave, the C1101 RF transceiver [5]. These 4 connections are a clock signal (SPI_SCLK), a serial communication channel from the MCU to transceiver (SPI_MOSI), a serial communication channel from the transceiver to MCU (SPI_MISO), and a slave select line (SPI_NSS). The transceiver subsystem has a few peripheral components which are required for proper functionality, which we define as wireless communication between the fob and VCU at a distance of at least 3 m.

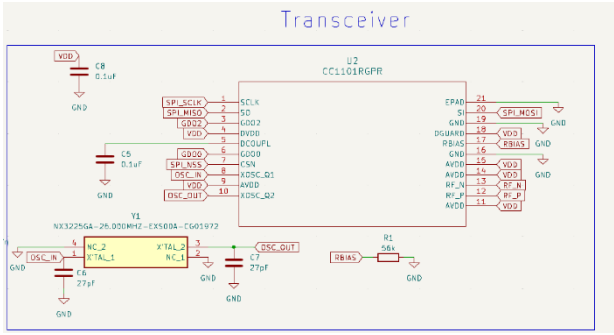


Figure 4 shows the Transceiver Circuit Schematic made within KiCad

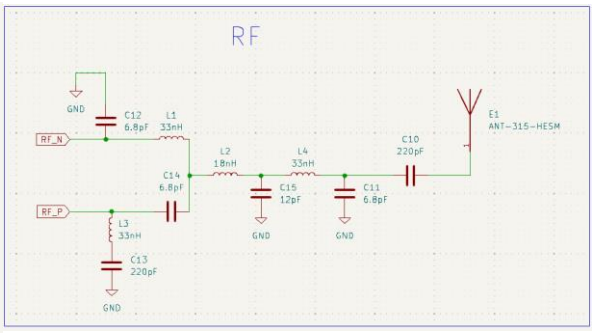


Figure 5 shows the RF chain Circuit Schematic made within KiCad

RF Transceiver/RF Chain R&V Table

Requirement	**Verification**	**Result**
Communication from at least 3m	Program one board to send a 1-byte message and the other to poll the number of bytes in the RX buffer.	Successfully verified. (See Figure 6)
Signal integrity from at least 3m	Send a message from one board to another, print received to UART, view on oscilloscope	Successfully verified. (See Figure 6)

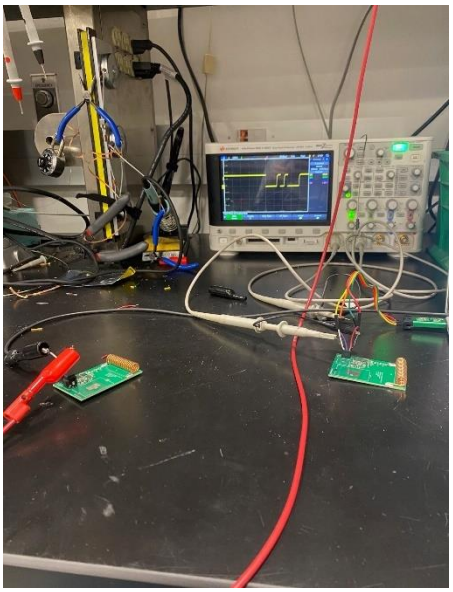


Figure 6. Concisely shows setup for verifying RF communication. Device on the left programmed to send a specific byte, received byte is displayed on an oscilloscope. Devices were separated by >3m and tested repeatedly.

Fob Power System

One of the largest bottlenecks when designing mobile devices is power. Thus, one of our most important subsystems is the power tree for our fob. The power subsystem consists of a 3V coin cell battery, a MAX16054 on/off controller IC [2] which will allow us to detect the noisy signal from the push button, power up the LDO [3] , MCU, transceiver, and begin the Schnorr Identification Protocol. 3V was chosen because it is the most common voltage for coin cell batteries and our MCU and RF transceiver can be powered by as little as 2V and 1.8V, respectively. We are using a push button as the input to the MAX16054 [2] (per the typical application circuit). The MAX16054 [2] output is connected to the enable pin on the 2.8V LDO [3] which powers the MCU and the transceiver. This topology is used because it will limit our quiescent current draw to the sum of the quiescent currents of the MAX16054 [2] and the LDO [3]. In the appendix, we provide a tolerance analysis showing that this power architecture allows a theoretical battery life of ~17 months.

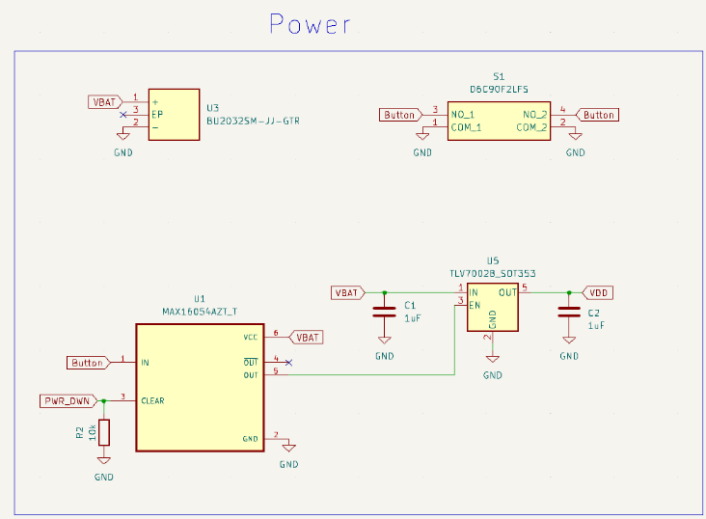


Figure 7. shows circuit schematic for power subsystem made within KiCad.

Fob Power R&V Table

Requirement	**Verification**	**Result**
Power down with on/off controller works correctly	Press the button while probing the LDO output, verify that after the CLEAR pin goes high on the MAX16054, we see the LDO output disabled (<0.001 V).	Successfully verified. (See figure 8)
Reverse voltage protection	Verify that it is impossible to insert a battery the wrong way into our holder, or if it can be jammed in, there is no output that reaches the MAX16054.	Successfully verified.
Over voltage protection	Make sure there does not exist a coin cell battery that is commonly available with over 5.5V, as this is the lower of the voltage maximums for the MAX16054 and the TLV70028 2.8V LDO	Successfully verified.

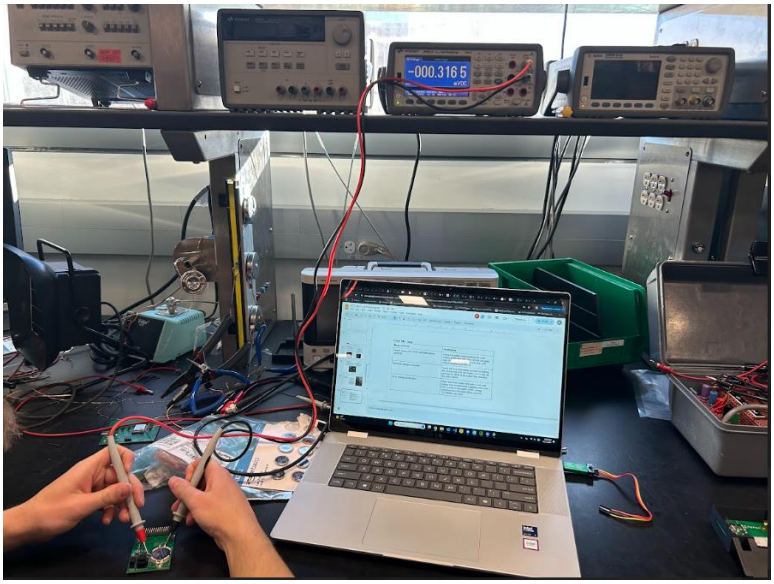


Figure 8. This shows the output of the LDO following an exchange to be -0.3 mV.

Fob MCU

We leveraged the SPI interface, UART interface, and a few GPIOs on the microcontroller. As explained in greater detail in the RF transceiver section, we used a full-duplex SPI interface to allow communication between the MCU and the RF transceiver. There are also a few output pins on the transceiver which were programmed as indicator pins (GDO0) which was used as an indication that a message had been received. One GPIO pin will be configured as an output which will send a high signal to the clear pin on the on/off controller when a transaction has been completed between the fob and the verification and control unit.

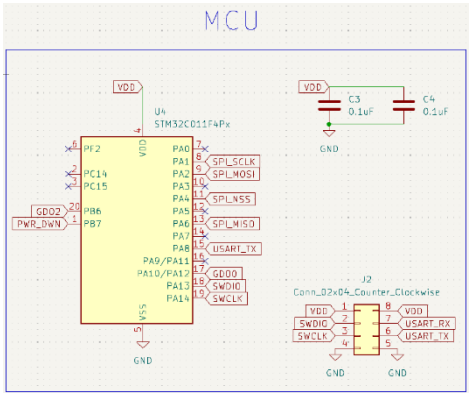


Figure 9 shows circuit schematic for Fob MCU made within KiCad.

Fob R&V Table

Requirements	Verification	Result
MCU can print statements to console via UART-USB converter	Write an MCU program to send a basic message to UART-USB converter to a laptop.	Successfully verified (See Figure 10)
Schnorr Identification Protocol code implementation can fit on MCU	Verify that the entire software can fit on the MCU Flash memory (16 KB)	Successfully verified

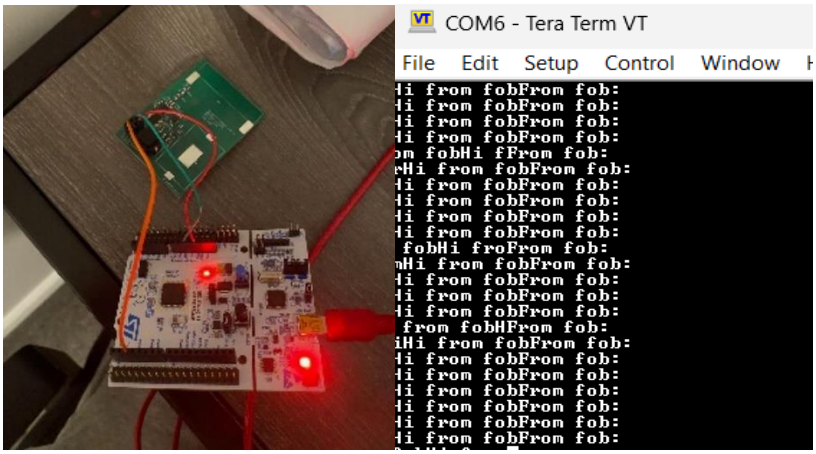


Figure 10. Left shows the setup using an STM32NUCLEO Dev Board as a UART-USB converter. Right shows the output on a laptop terminal.

Verification and Control Unit (VCU) MCU

This MCU is very similar to the MCU on the FOB side where we communicate between the MCU and the transceiver. The main difference between the two is there are a couple of extra output pins being used on the Verification side. PB6 and PB7 both represent the motor control: when PB6 is high and PB7 is low the motor will go left, when vice versa the motor will go right, and when both are high the motor will stop. Additionally, we have PA3, PA5, and PA11 connected to LEDs each representing the different states of our VCU: transmitting a signal, receiving a signal, and when a valid unlock sequence has taken place.

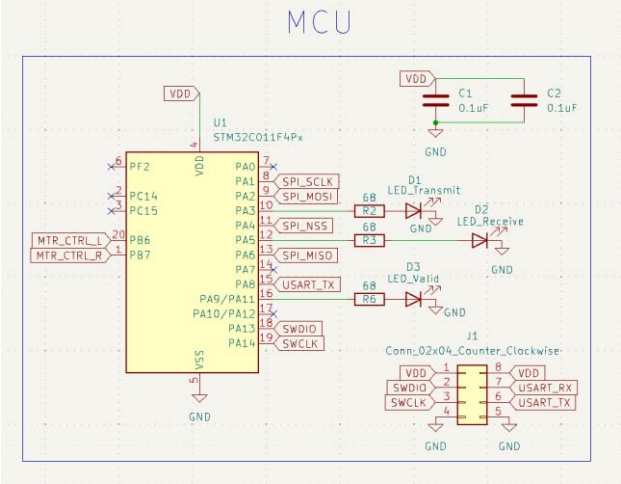


Figure 11 shows the circuit schematic for Verification and Control Unit's MCU made within KiCad.

MCU R&V Table

Requirements	**Verification**	**Result**
MCU can receive and transmit signals	Verify proper indicator LEDs light up, view received packets on oscilloscope (probe UART TX)	Successfully verified (See Figure 12)
Schnorr Identification Protocol code implementation can fit on MCU	Verify that the entire software program can fit on the MCU in flash memory (16 KB). If our program is able to compile and run, then this will be verified.	Successfully verified (See Figure 12)

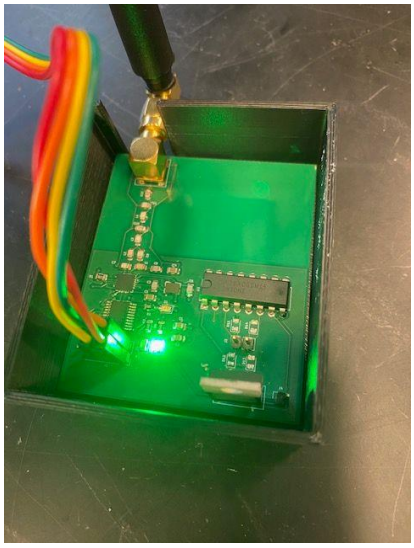


Figure 12 shows a green LED on the VCU is lit indicating a valid unlock sequence.

Verification and Control Unit (VCU) Power

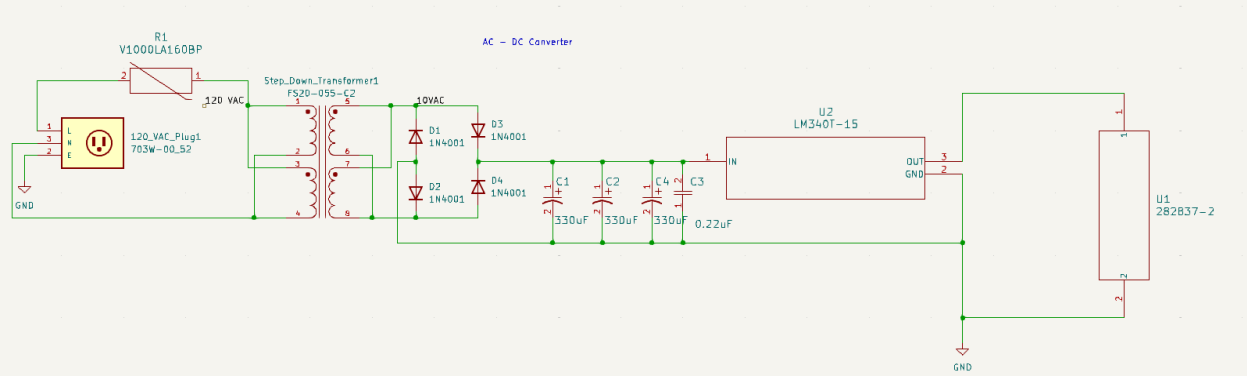


Figure 13 shown above displays the AC-DC power converter as a circuit schematic, utilizing different components for stepping down voltage, Rectification, Filtering, and Regulation made within KiCad.

The power subsystem for our VCU is very different to our power subsystem on our key FOB. We decided to design our own AC to DC converter, based on the final project of ECE 343, using 4 1N4001 diodes [6], a 10V to 5V linear regulator [7], and capacitors, along with a 5V to 3.3V LDO regulator [8] to provide the power to our VCU. We know this design works as it worked for our final project from ECE 343.

AC-DC Converter and Regulator Table

Requirements	Verification	Results
AD-DC converter converts 120 VAC to around 5 VDC AC-DC converter will also output a value of around 1.5 amps to power the motor and other subcomponents.	Testing consists of verifying signal steps down, rectified, filtered, and regulated for a smoothed out 5 VDC output using an oscilloscope and digital multimeter	Failed stepping down voltage from 120 VAC to 10 VAC Successfully verified converter rectified, filtered, and regulated 10 VAC to 5 VDC.
5 to 3.3 voltage regulator converts voltage properly.	Test using an oscilloscope to ensure that a 5V in gets converted to a 3.3V out.	Successfully verified (See figure 14).

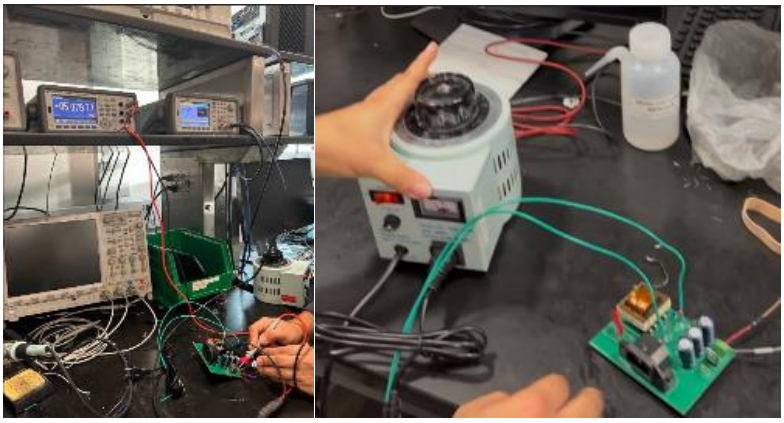


Figure 14 displays work-around solution for the faulty transformer and connector consisting of a VARIAC Meter, along with testing to prove 10 VAC from a signal generator gets converted to 5 VDC

Verification and Control Unit (VCU) Motor

Our final subsystem for our VCU is our Motor and Motor Driver. The whole purpose of our motor subsystem is to show a physical output to the unlocking process. We are using a basic 5 V BLDC motor and an L293D as our motor driver. Our motor driver will take input from our MCU to control the direction of the motor spin.

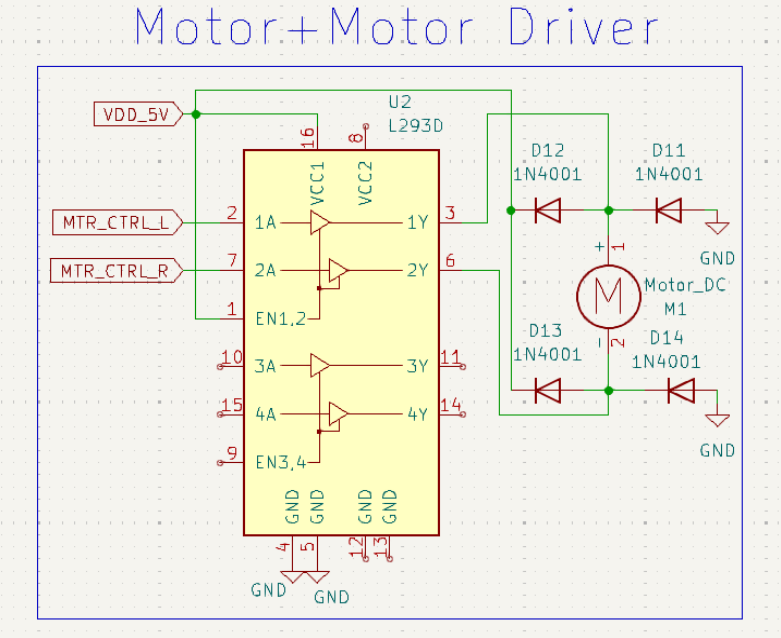


Figure 15 above shows our the circuit schematic of our motor driver and motor [9] made within KiCad

Motor and Motor Driver Table

Requirements	**Verification**	**Result**
Motor can spin either in the clockwise or counterclockwise direction during the duration at which it's receiving a high signal. If pin 1 receives a high, it will turn clockwise, and if pin 2 receives high it will turn counterclockwise.	Will test sending High to both the 1A and 2A pins of the motor driver to ensure the motor subsystem works. Note that if both pins receive high or low, the motor will not spin, due to no potential	Successfully verified (See Figure 16)

	difference across the motor.	

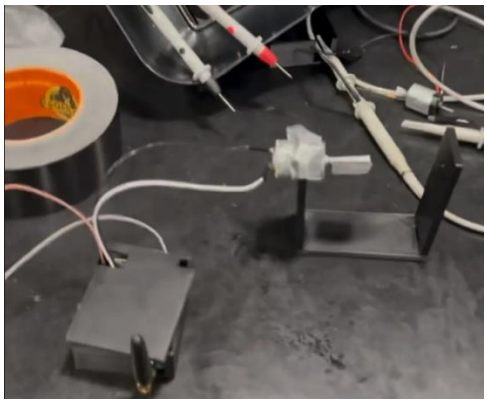
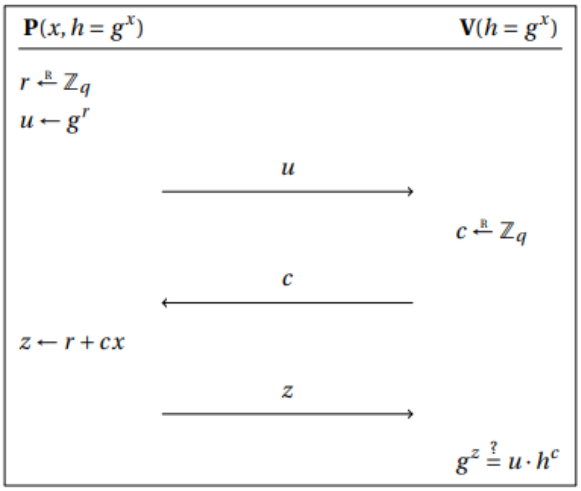


Figure 16 above shows an image taken from our demo video of our motor spinning after a valid unlock sequence.

Software

2 Schnorr's Protocol: Proof of Knowledge of Discrete Log

Suppose that a prover wants to prove it knows the discrete logarithm x of some group element $h = g^x \in \mathbb{G}$, where $\mathbb{G}$ is a group of prime order q. Here $\mathcal{R} = \{(x, h) \in \mathbb{Z}_q \times \mathbb{G} : g^x = h\}$, where the group $\mathbb{G}$ and the generator g are public parameters.



Completeness: if $z = r + cx$, then $g^z = g^{r+cx} = g^r \cdot (g^x)^c = u \cdot h^c$.

Figure 17 above shows the Schnorr protocol sequence [1]

Fob Software

The software which will be run on the Fob MCU will take on the role of the prover (P) in the Schnorr identification protocol. This means that the Fob will send the initial "open" request, receive a "challenge" wirelessly from the verification and control unit, and then calculate the correct response to the challenge and send it back to the verification and control unit. The fob will be loaded with a public key, $h=g^x$, private key, x, which are both 256-bit numbers, during the initial software flash.

Note that g is the generator of a modular group G. An example would be group G which has numbers modulo 7. An example is shown below.

The group G consists of the non-zero integers in $\mathbb{Z}_7$, i.e., $G = \{1, 2, 3, 4, 5, 6\}$.

We choose $g = 3$ as the generator. Now, let's compute the powers of g mod 7:

$$
\begin{aligned}
g^1 &= 3^1 \mod 7 = 3, \\
g^2 &= 3^2 \mod 7 = 9 \mod 7 = 2, \\
g^3 &= 3^3 \mod 7 = 27 \mod 7 = 6, \\
g^4 &= 3^4 \mod 7 = 81 \mod 7 = 4, \\
g^5 &= 3^5 \mod 7 = 243 \mod 7 = 5, \\
g^6 &= 3^6 \mod 7 = 729 \mod 7 = 1.
\end{aligned}
$$

Thus, the powers of $g = 3 \mod 7$ generate the set $\{1, 2, 3, 4, 5, 6\}$, which means $g = 3$ is a generator of the group $G \mod 7$.

Figure 18 above shows further explanation of Schnorr [1]

Modular Group G, Generator g Example

When the button is pressed, the fob MCU will generate a random number *r* within the group *G* and send $u=g^r$ to the verification and control unit. After receiving the challenge *c* from the verification and control unit, the fob will compute $z=r + cx$. That computation requires knowledge of the private key *x*. The fob will then send *z* to the verification and control unit. If z was correctly computed, then $g^z=u*h^c$. Since *g* and *h* are preprogrammed on the verification and control unit, both sides of the equation can be evaluated and if the equation holds, then the unlock command is coming from a fob with an authorized public key.

Verification and Control Unit Software

The verification and control unit will be the verifier (V) in the Schnorr Identification Protocol. After receiving the initial "open" command from the fob, we will store it for the final verification calculation. The MCU will generate a random number in group *G* and send it to the fob as a challenge *c*. The verification and control unit will then wait for the response *z* from the fob which it will store and then begin the verification calculation to see if $g^z=u*h^c$. If yes, the verification and control unit will illuminate the "Valid Exchange" LED and then send a signal to the L293D motor driver IC [9] to spin the motor, simulating a garage door opening or car unlocking.

Costs

Team member compensation

$40 * 2.5 * 50 = $5000 each

$5000 * 3 members = $15000

Fob Module		
Description	**Manufacturer**	**Cost**
RF ANT 315MHZ HELICAL SOLDER SMD	TE Connectivity Linx	$1.18
SWITCH PUSH SPST-NO 0.1A 32V	C&K	$1.29
IC RF TXRX ISM<1GHZ 20QFN	Analog Devices Inc./Maxim Integrated	$4.30
IC RF TXRX ISM<1GHZ 20QFN	Texas Instruments	$3.63
BATTERY HOLDER COIN 20MM SMD	MPD (Memory Protection Devices)	$1.24
IC MCU 32BIT 16KB FLASH 20TSSOP	STMicroelectronics	$1.71
IC REG LINEAR 2.8V 200MA SC70-5	Texas Instruments	$0.15
CRYSTAL 26.0000MHZ 10PF SMD	NDK America, Inc.	$0.66
Total		**$14.16**

VCU Module		
Description	**Manufacturer**	**Cost**
R280 3-6V 12000RPM BRSHD DC MTR	OSEPP Electronics LTD	$4.45
IC MCU 32BIT 16KB FLASH 20TSSOP	STMicroelectronics	$1.71
16 Pin Motor Driver IC, 4.5-36V @1A, L293D	Texas Instruments	$2.95

VCU Module		
IC RF TXRX ISM<1GHZ 20QFN	Texas Instruments	$3.63
IC REG LINEAR 3.3V 800MA TO220-3	Texas Instruments	$1.79
CRYSTAL 26.0000MHZ 10PF SMD	NDK America, Inc.	$0.66
Total		**$15.19**

Total Parts Cost= $15.19+$14.16 = $29.35

Assume a 5% shipping cost: Shipping=$29.35*5% = $1.47

Assume 10% sales tax: Tax = $29.35 *10% = $2.94

Final Total Cost = Team Member Compensation + Total Parts Cost + Shipping + Tax

Final Total Cost = $15,000+$29.35+$1.47+$2.94= $15,033.76

Conclusion

Our project met all objectives and demonstrated functionality. We successfully achieved accurate wireless communication over a range of 3 meters, motor actuation within 2 seconds of sending the unlock signal, and resilience against replay attacks. By translating a theoretical cryptographic exchange into a practical hardware implementation, we integrated secure authentication with real-world system control.

Our shortcomings did not impact basic functionality, and only limited the full expression of our platform's capabilities. Our implementation of the Schnorr Identification Protocol used a reduced security parameter (key length) due to memory limitations on the microcontrollers. A minor bug which prevents 2 consecutive unlock signals from being received, regardless of timing, could presumably be resolved by leveraging a status pin on the transceiver instead of a continuous FIFO read.

Improvements to the Fob design would involve size reduction by using smaller footprint components and a more efficient layout of the RF chain. We did not prioritize testing different antennas but using a smaller antenna or even a PCB trace antenna is likely the largest source of possible size reduction. Using a microcontroller with more flash memory would allow us to use 256-bit keys for our Schnorr Identification Protocol implementation and allow us to sample the ADC as a source of pseudo-random number generation. Size reduction is less of a priority for a VCU redesign, and the current antenna should be kept

ensuring signal strength. An improvement that would be suggested in commercialization would be to use an off-the-shelf AC-DC converter and use a barrel jack connector or something comparable.

Our initial vision of a cryptographically secure remote access system took the form of a garage door opener, and the first addressable market may be the retrofitting of insecure garage door models. However, we see Schnorr Identification Protocol as a promising encryption scheme for remote entry systems.

Ethics and Safety

The nature of any project dealing with security requires research into current attacks and the vulnerabilities of current systems. This requires us to gain the knowledge that would be required to perform these attacks. As ethical engineers, we will not use this information for malicious reasons. One compliance concern we have is the RF power of the transceiver, we ensure that our transceiver adheres to the FCC requirements: 787 kHz bandwidth and 200 uV/m at 3 m distance.

Since we are using an AC-DC converter designed to be used with a standard wall outlet, we needed to make sure our converter properly converts the 120V AC, typical for wall outlets, to 5V, the desired voltage for our garage door lock to function properly. One specific safety concern is over current, which is protected with a fuse.

Citations

[1] Dima Kogan (2019), “Lecture 5: Proofs of Knowledge, Schnorr’s protocol, NIZK”, CS355, Computer Science Department, Stanford University [Online]. Available:

https://crypto.stanford.edu/cs355/19sp/lec5.pdf

[Accessed: Sept. 30, 2024]

[2] Maxim Integrated Products, “On/Off Controller with Debounce and ±15kV ESD Protection”, Rev 0, May 2008 [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX16054.pdf

[3] Texas Instruments, “200-mA, Low-IQ, Low-Dropout Regulator (LDO) for Portable Devices”, Rev C, June 2018 [Online]. Available:

https://www.ti.com/lit/ds/symlink/tlv700xx-q1.pdf?ts=1727978597283&ref_url=https%253A%252F%252Fwww.google.com%252F

[4] STMicroelectronics, “STM32C011x4/x6”, Rev. 4, Jan. 2024 [Online]. Available:

https://www.st.com/en/microcontrollers-microprocessors/stm32c011f4.html

[5] Texas Instruments, “Low-Power Sub-1 GHz RF Transceiver”, Revision SWRS061I, Nov. 2013 [Online]. Available:

https://www.ti.com/lit/ds/symlink/cc1101.pdf?ts=1727939599209&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC1101

[6] Onsemi, “Axial-Lead Glass Passivated Standard Recovery Rectifiers”, Rev.18, June 2024 [Online]. Available:

https://www.onsemi.com/pdf/datasheet/1n4001-d.pdf

[7] Texas Instruments, “LM340, LM340A and LM7805 Family Wide VIN 1.5-A Fixed Voltage Regulators”, Rev D, Sept. 2016 [Online]. Available:

https://www.ti.com/lit/ds/symlink/lm340.pdf?ts=1733946012496&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLM340%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dapp-lp-null-44700045336317407_prodfolderdynamic-cpc-pf-google-ww_en_int%2526utm_content%253Dprodfolddynamic%2526ds_k%253DDYNAMIC+SEARCH+ADS%2526DCM%253Dyes%2526gad_source%253D1%2526gclid%253DCjwKCAiAjeW6BhBAEiwAdKltMo48yHRZmMv84pvzUJ4CI9WNawPxeC8Swh2ab4e38PniP25A01HENhoCvsYQAvD_BwE%2526gclsrc%253Daw.ds

[8] Texas Instruments, “LM3940 1-A Low-Dropout Regulator for 5-V to 3.3-V Conversion”, Rev. G, Feb. 2015 [Online]. Available:

https://www.ti.com/general/docs/suppproductinfo.tsp?distId=10&gotoUrl=https%3A%2F%2Fwww.ti.com%2Flit%2Fgpn%2Flm3940

[9] Texas Instruments, “L293x Quadruple Half-H Drivers “, Rev.D, Jan. 2016 [Online]. Available:

https://www.ti.com/lit/ds/symlink/l293d.pdf?ts=1727938192243&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FL293D

Appendix A

A.1 Diagrams from CC1101 Datasheet

A.2 Diagram from MAX16054 Datasheet

A.3 Diagrams from STM32CubeIDE

A.4 Housing CAD Renderings

A.5 Tolerance Analysis

A.1 Diagrams from CC1101 Datasheet (Citation 5)

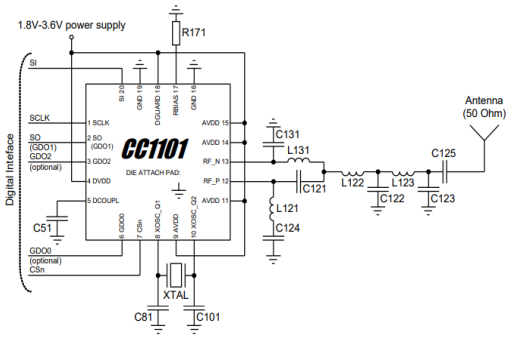


Figure 10: Typical Application and Evaluation Circuit 315/433 MHz (excluding supply decoupling capacitors)

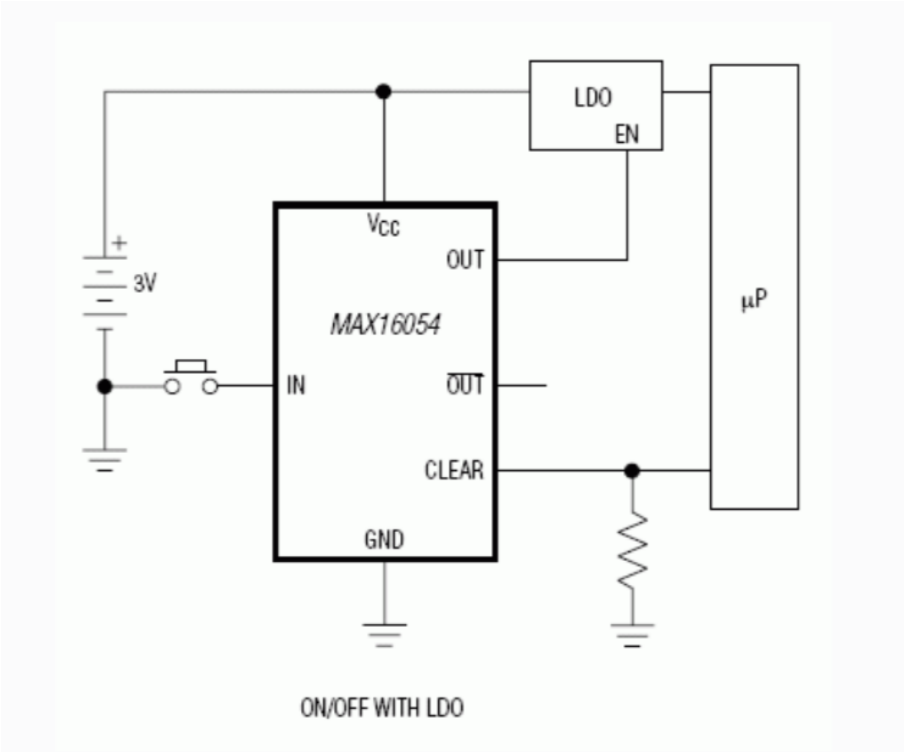
Component	Description
C51	Decoupling capacitor for on-chip voltage regulator to digital part
C81/C101	Crystal loading capacitors
C121/C131	RF balun/matching capacitors
C122	RF LC filter/matching filter capacitor (315/433 MHz). RF balun/matching capacitor (868/915 MHz).
C123	RF LC filter/matching capacitor
C124	RF balun DC blocking capacitor
C125	RF LC filter DC blocking capacitor and part of optional RF LC filter (868/915 MHz)
C126	Part of optional RF LC filter and DC-block (868/915 MHz)
L121/L131	RF balun/matching inductors (inexpensive multi-layer type)
L122	RF LC filter/matching filter inductor (315 and 433 MHz). RF balun/matching inductor (868/915 MHz). (inexpensive multi-layer type)
L123	RF LC filter/matching filter inductor (inexpensive multi-layer type)
L124	RF LC filter/matching filter inductor (inexpensive multi-layer type)
L125	Optional RF LC filter/matching filter inductor (inexpensive multi-layer type) (868/915 MHz)
L132	RF balun/matching inductor. (inexpensive multi-layer type)
R171	Resistor for internal bias current reference
XTAL	26 – 27 MHz crystal

Table 20: Overview of External Components (excluding supply decoupling capacitors)

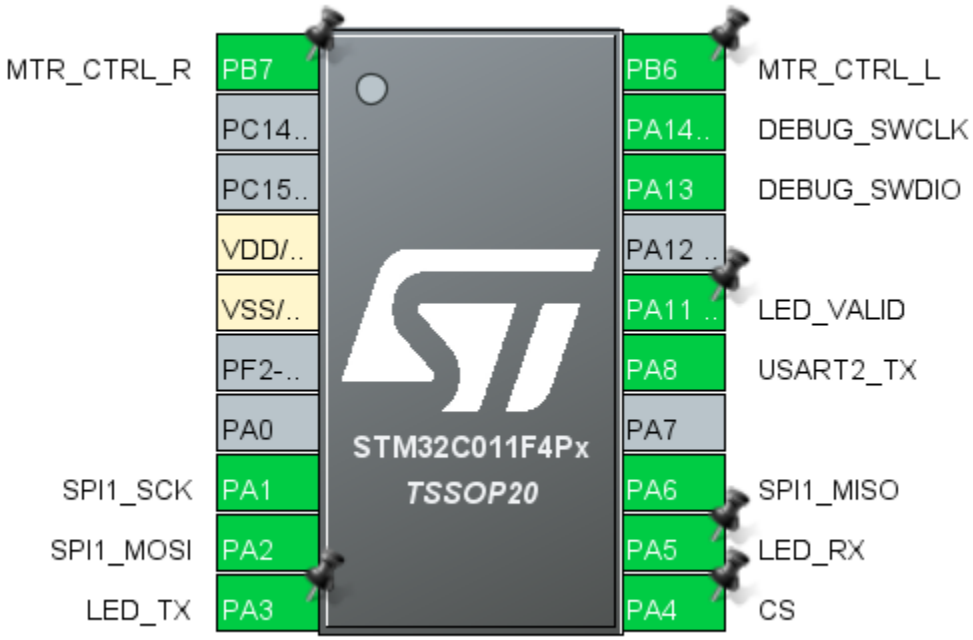
Component	Value at 315MHz	Value at 433MHz	Value at 868/915MHz	Manufacturer
C51	100 nF ± 10%, 0402 X5R			Murata GRM1555C series
C81	27 pF ± 5%, 0402 NP0			Murata GRM1555C series
C101	27 pF ± 5%, 0402 NP0			Murata GRM1555C series
C121	6.8 pF ± 0.5 pF, 0402 NP0	3.9 pF ± 0.25 pF, 0402 NP0	1.0 pF ± 0.25 pF, 0402 NP0	Murata GRM1555C series
C122	12 pF ± 5%, 0402 NP0	8.2 pF ± 0.5 pF, 0402 NP0	1.5 pF ± 0.25 pF, 0402 NP0	Murata GRM1555C series
C123	6.8 pF ± 0.5 pF, 0402 NP0	5.6 pF ± 0.5 pF, 0402 NP0	3.3 pF ± 0.25 pF, 0402 NP0	Murata GRM1555C series
C124	220 pF ± 5%, 0402 NP0	220 pF ± 5%, 0402 NP0	100 pF ± 5%, 0402 NP0	Murata GRM1555C series
C125	220 pF ± 5%, 0402 NP0	220 pF ± 5%, 0402 NP0	12 pF ± 5%, 0402 NP0	Murata GRM1555C series
C126			47 pF ± 5%, 0402 NP0	Murata GRM1555C series
C131	6.8 pF ± 0.5 pF, 0402 NP0	3.9 pF ± 0.25 pF, 0402 NP0	1.5 pF ± 0.25 pF, 0402 NP0	Murata GRM1555C series
L121	33 nH ± 5%, 0402 monolithic	27 nH ± 5%, 0402 monolithic	12 nH ± 5%, 0402 monolithic	Murata LQG15HS series (315/433 MHz) Murata LQW15xx series (868/915 MHz)
L122	18 nH ± 5%, 0402 monolithic	22 nH ± 5%, 0402 monolithic	18 nH ± 5%, 0402 monolithic	Murata LQG15HS series (315/433 MHz) Murata LQW15xx series (868/915 MHz)
L123	33 nH ± 5%, 0402 monolithic	27 nH ± 5%, 0402 monolithic	12 nH ± 5%, 0402 monolithic	Murata LQG15HS series (315/433 MHz) Murata LQW15xx series (868/915 MHz)
L124			12 nH ± 5%, 0402 monolithic	Murata LQG15HS series (315/433 MHz) Murata LQW15xx series (868/915 MHz)
L125			3.3 nH ± 5%, 0402 monolithic	Murata LQG15HS series (315/433 MHz) Murata LQW15xx series (868/915 MHz)
L131	33 nH ± 5%, 0402 monolithic	27 nH ± 5%, 0402 monolithic	12 nH ± 5%, 0402 monolithic	Murata LQG15HS series (315/433 MHz) Murata LQW15xx series (868/915 MHz)
L132			18 nH ± 5%, 0402 monolithic	Murata LQG15HS series (315/433 MHz) Murata LQW15xx series (868/915 MHz)
R171	56 kΩ ± 1%, 0402	Koa RK73 series		
XTAL	26.0 MHz surface mount crystal			NDK, NX3225GA or AT-41CD2

Table 21: Bill Of Materials for the Application Circuit[1]

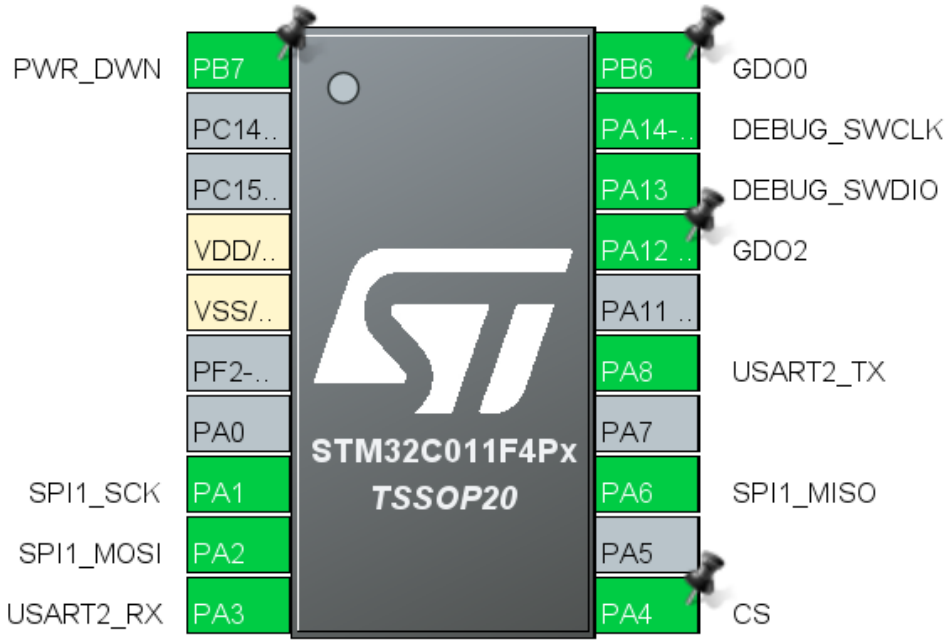
A.2 Diagram from MAX16054 Datasheet (Citation 2)



A.3 Diagrams from STM32CubeIDE

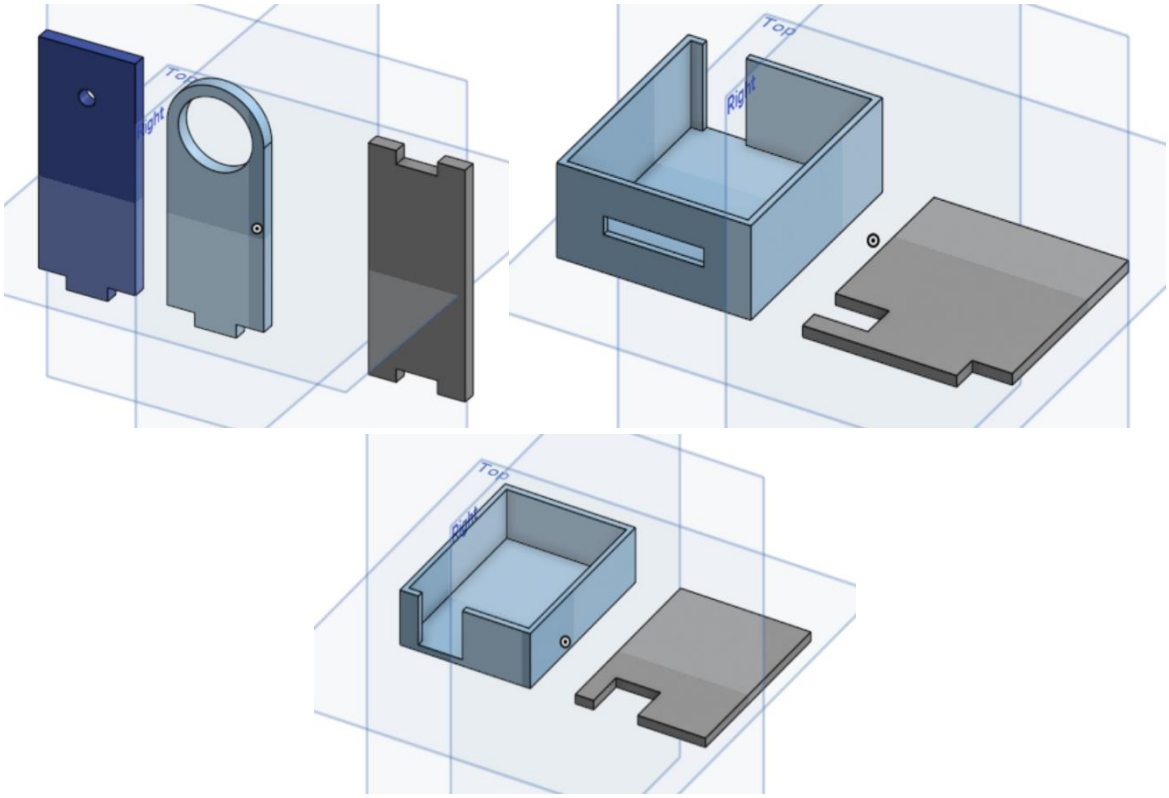


VCU



Fob

A.4 Housing CAD Renderings



A.5 Tolerance Analysis – Battery Life

1 Current Consumption When Off (1 month)

- The latch IC (MAX16054) consumes $7\,\mu A$.
- The LDO consumes less than $1\,\mu A$.
- 1 month is 720 hours, we will ignore the fact that during some of this duration the device will be active
- $I_{off} = 8\mu A \times \frac{1mA}{1000\mu A} \times 720\,\text{hours/month} = \boxed{5.75 \quad \text{mAh/month}}$

2 Active Current Consumption Per Interaction

- The latch IC (MAX16054) consumes $0.4\,mA$.
- The MCU consumes at most $4\,mA$ for running code from flash memory with 48 MHz clock, SPI requires $0.2\,mA$, the GPIOs will require $0.1\,mA$ each and there are 4.
- The LDO efficiency is calculated as $\frac{V_{out}}{V_{in}} = \frac{2.8}{3} = 0.93$. This means that the current requirement for the MCU and RF transceiver combined should be multiplied by 1.075.
- The RF transceiver consumes $14.7\,mA$ in RX mode and $30\,mA$ in TX mode.

We estimate the total time of operation for one interaction to be 2 seconds, or 0.0006 hours. This consists of:

- 0.6 seconds (0.0002 hours) in RX mode.
- 1.4 seconds (0.0004 hours) in TX mode.

The total current consumption per interaction is:

$$I_{\text{latch}} = 0.4\,\text{mA} \times 0.0006\,\text{hours} = 0.00024\,\text{mAh}$$

$$I_{\text{MCU}} = 1.075((4 + 0.2 + 0.4)\,\text{mA} \times 0.0006\,\text{hours}) = 0.002967\,\text{mAh}$$

$$\begin{aligned} I_{\text{transceiver}} &= 1.075 \times ((14.7\,\text{mA} \times 0.0002\,\text{hours}) + (30\,\text{mA} \times 0.0004\,\text{hours})) \\ &= 0.00294 + 0.012 = 0.0160605\,\text{mAh} \end{aligned}$$

$$I_{\text{interaction}} = I_{\text{latch}} + I_{\text{MCU}} + I_{\text{transceiver}}$$

$$I_{\text{interaction}} = 0.0192675\,\text{mAh} \approx \boxed{0.0193 \quad \text{mAh/interaction}}$$

3 Monthly Consumption

Assuming 10 interactions per day, the monthly current consumption is:

$$I_{\text{month}} = (I_{\text{off}}) + (10\,\text{interactions/day} \times 30\,\text{days/month} \times I_{\text{interaction}})$$

$$= 5.75\,\text{mAh/month} + 5.79\,\text{mAh/month}$$

$$I_{\text{month}} = \boxed{11.54 \quad \text{mAh/month}}$$

4 Battery Life Estimation

Our target is for the fob to last at least one month on a single battery. We have identified batteries with a capacity greater than 200 mAh. Therefore, the estimated battery life is at least:

$$t_{\text{life}} = \frac{200\,\text{mAh}}{11.54\,\text{mAh/month}} \approx \boxed{17 \quad \text{months}}$$

This is 17x longer than our goal