Homework 6: Modeling
Team 12

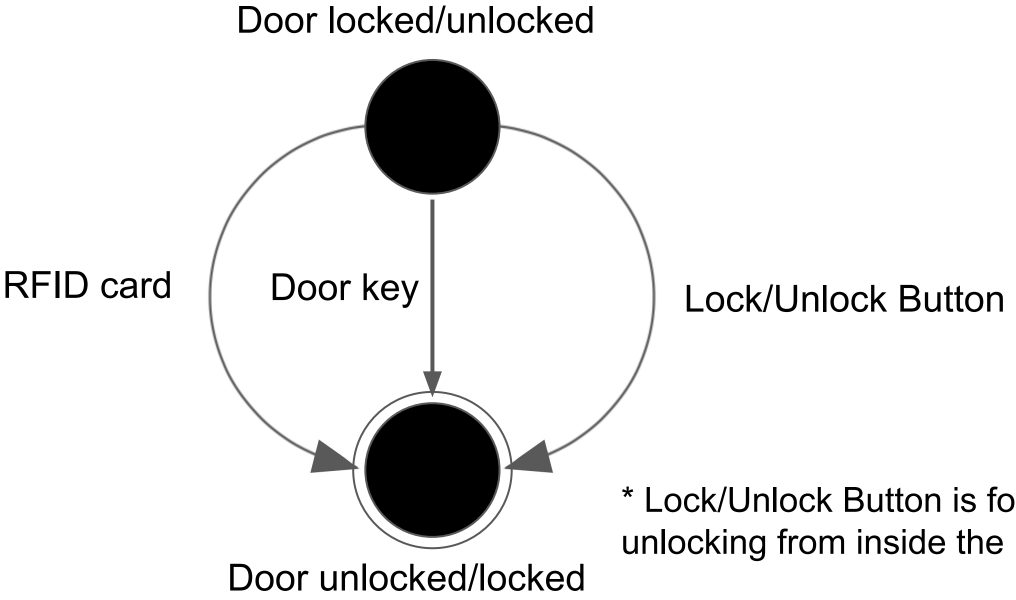
Version 2.0

December 8, 2020

Miguel Higgins Moy, Miles Breslin
Noah Page, John Sharpe

State Diagram

Clearly describes system states, and inputs to change states. Not that helpful since there are only two states and three inputs, but easy to understand.

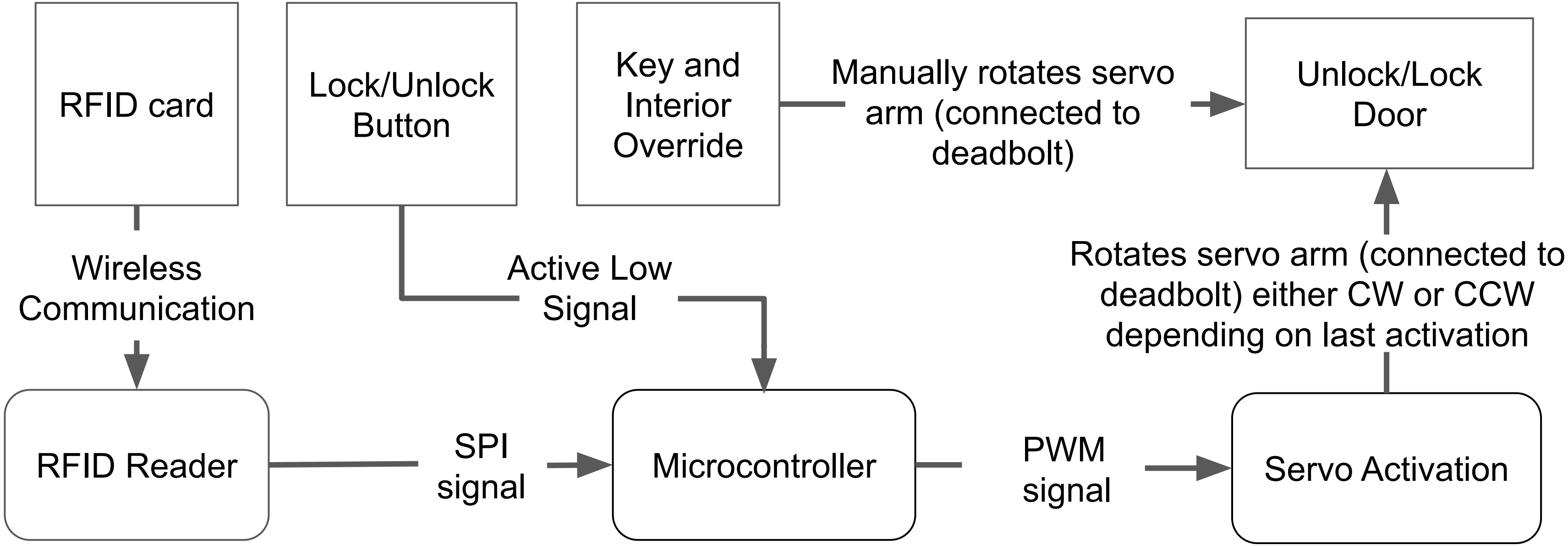


* Lock/Unlock Button is for locking/ unlocking from inside the house

Data Flow Diagram Notes

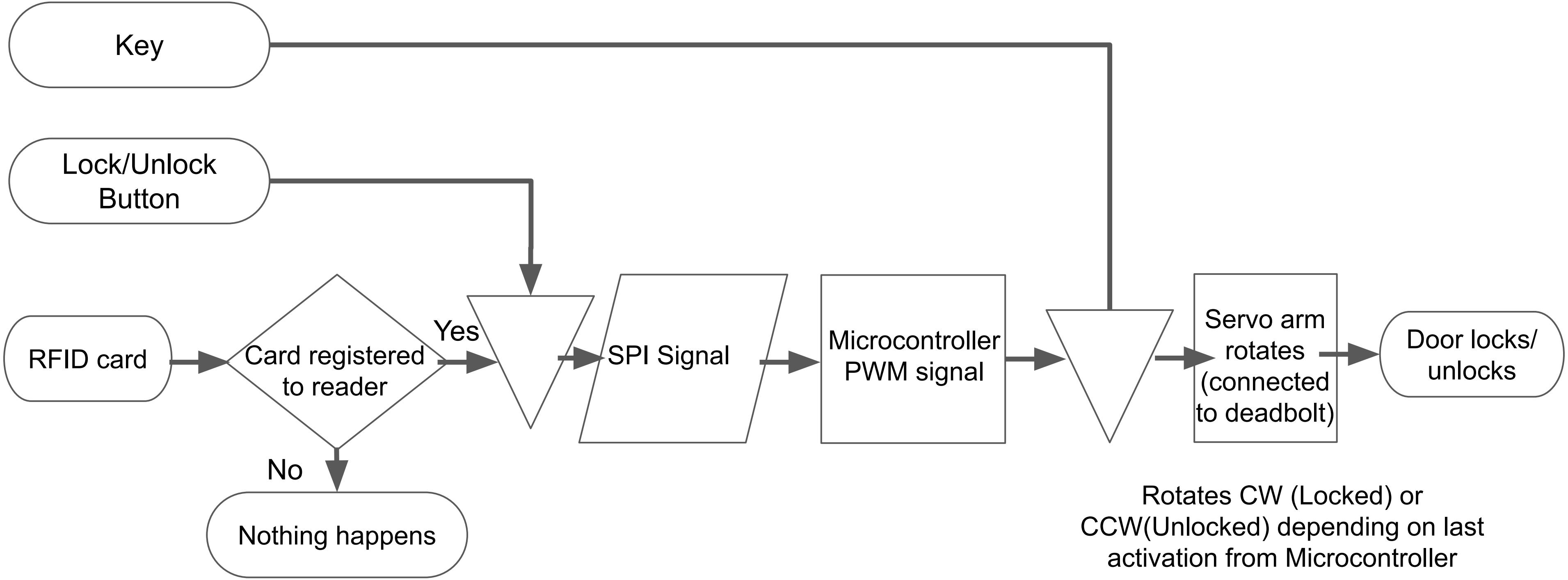
- **Key/RFID cards** are interfaces. They are external inputs to the system.
- The **RFID reader** is a process since it does not store the state of the door (locked/unlocked), it just interprets an external signal.
- The **microcontroller** is a process since it poles the position of the servo arm to send the correct signal needed to lock/unlock the door. The processor cannot store the data directly because the door can also be locked/unlocked with a key, which bypasses the processor.
- The **servo** arm location depends on the state of the door. If the normal key is used, the arm position follows the door. For that reason the servo poles the door state, making it a process.
- The state of the lock (locked/unlocked) is a data storage system.

Data Flow Diagrams (DFDs)


RFID card
Lock/Unlock Button
Key and Interior Override
Manually rotates servo arm (connected to deadbolt)
Unlock/Lock Door
Wireless Communication
Active Low Signal
Rotates servo arm (connected to deadbolt) either CW or CCW depending on last activation
RFID Reader
SPI signal
Microcontroller
PWM signal
Servo Activation

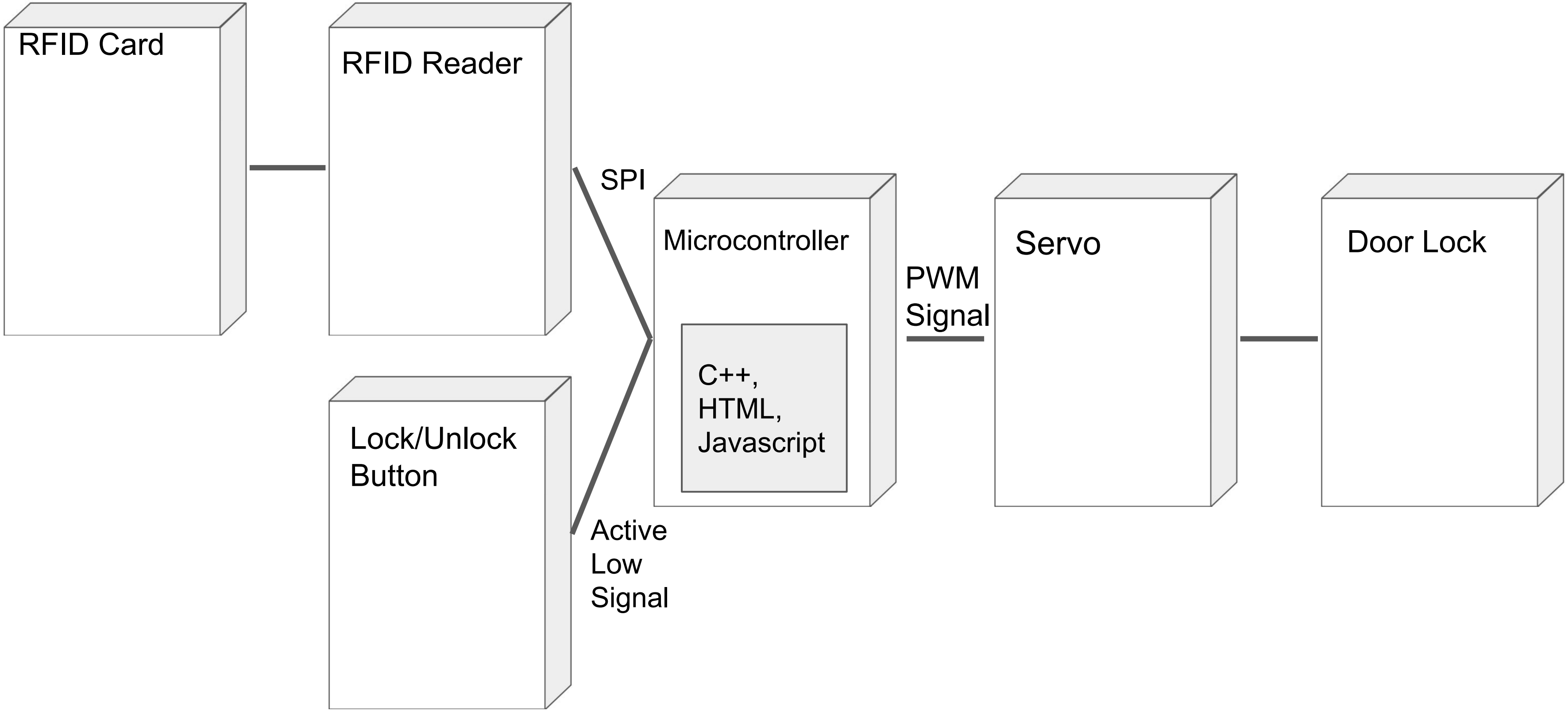
* Lock/Unlock Button is for locking/unlocking door from inside the house

Flowcharts


Key
Lock/Unlock Button
RFID card
Card registered to reader
Yes
No
Nothing happens
SPI Signal
Microcontroller PWM signal
Servo arm rotates (connected to deadbolt)
Door locks/ unlocks

Rotates CW (Locked) or CCW(Unlocked) depending on last activation from Microcontroller

UML Physical View



Rejected Model Types

- Data Flow Diagram Table:
  - There are only two possible events at both lead to the same outcome. Since our model does not have many components, the data flow diagram is likely easier to follow than the table.
- Entity Relationship Diagrams
  - This would be more helpful for classifying data types or objects with multiple attributes. Since our system is linear, this diagram would not convey much information.
- UML Static View:
  - Great for programming, explains how to access a program/class/struct and its attributes. Since our system is linear, almost all components have one input one output, which is better represented with a Data Flow Diagram or Flowchart.
- UML Activity View
  - Great for showing a parallel process and more high level approach. Otherwise, a Data Flow Diagram or Flowchart offers more specific information through the different block shapes.
- UML Interaction View
  - Shows interaction between blocks, great when a block has multiple inputs/outputs, but not very useful in our project, where most blocks have one input one output.