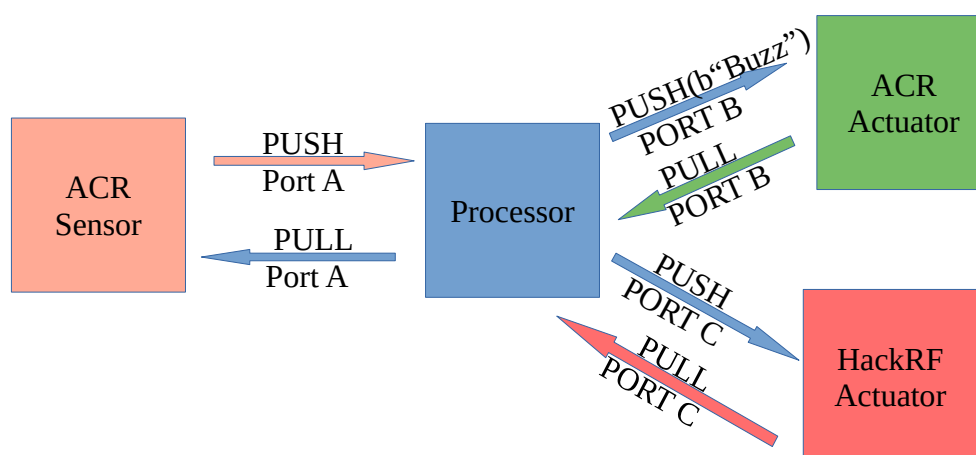


For Homework 12, we will make a smart system using the tools and technologies that we have previously encountered in the course.

How the system works [5 pts]

- Each time a card is placed on the ACR1252 reader (the sensor) it reads the card's UID and passes it to the processing unit (your computer and a python script)
 - The date and time when the UID was detected is stored in a log file along with the card type (e.g. Ultralight C, MIFARE Classic etc.)
 - File name: *nfc_sensor.log*
- When the processing unit receives a UID, it checks if it is on an approved list (this can be a list of the UIDs of the cards in your possession – one card in the list is also fine)
- If the UID is approved and it is between the hours of 09:00 and 17:00 then the doorbell should ring (*i.e. start ringing at 09:00 and end at 16:59:59*)
 - Each time the doorbell is rung, the date and time when the event occurred is written to a log file
 - File name: *doorbell.log*
- If the UID is approved but it is between 17:00 and 08:59, the card reader LED should blink orange 5 times with the buzzer off. (*i.e. start flashing from 17:00 and end at 08:59:59*)
 - The LED 'on time' should be 4 times longer than the off time.
 - Each time this happens, the date and time should be stored in a log file
 - File name: *reader.log*
- If the UID is not approved, the card reader buzzer should give 1 long beep with the LED off
 - Each time this happens, the date and time should be stored in a log file
 - File name: *reader.log*
- For communication between the components:
 - ZMQ PUSH/PULL method should be used:
 - Use a different port for communication between the processor and the other components
 - For the PUSH messages, use words to indicate to the actuator/processor the action that has to be completed e.g. "Buzz" can indicate to the ACR that it needs to sound the buzzer. This way we can push many commands to a device (e.g. "UID", "ATR")
 - See diagram for illustration of configuration settings:



- The different ports are used to prevent the error:
 - `zmq.error.ZMQError: Address already in use`
- For the log files
 - The current date should be entered first in the format yyyy/mm/dd
 - e.g. 2023/11/15

- If the current date is already in the file do not add it again
- Every event entry should be added on a new line with the time
 - Examples:
 - For `nfc_sensor.log`

```
2023/11/08
    05:15:10      MIFARE Classic 1K detected: UID 01234567
    16:40:45      Unknown detected: UID ABCDEF01

2023/11/16
    18:35:22      MIFARE Ultralight C detected: UID 0123456789ABCD
```
 - For `doorbell.log`

```
2023/11/07
    10:12:21      Doorbell Ring Failed

2023/11/16
    16:40:52      Doorbell Ring Success
```
 - For `reader.log`

```
2023/11/08
    05:15:10      Orange LED Blink Success
    18:40:45      Orange LED Blink Failed

2023/11/16
    18:35:22      Buzzer Success
```

- For the sensor, the functions and list in `card_identifier.py` can be used to identify the card.
 - Try testing with your bank card to see how it handles unknown ATRs
- For the reader, the functions and examples in `acr1252_actions.py` can be used to help control the LEDs and buzzer. Remember to set the correct `r[?].createConnection()` index.
- For the doorbell, use a replay attack to ring it, either with the `hackrf_transfer` command (*preferred as it is less likely to hang*) or by running the python script. The *configure* doorbell function(s) can be used to capture a door bell signal and the *ring doorbell* function(s) can be used to replay the captured signal.
- Template files have been provided for each component: processor (the computer), UID sensor (the ACR), doorbell actuator (the HackRF) and buzzer and LED actuator (the ACR).
- Sample logs are available with the template files.

Bonus [3 pts]

Using MQTT and Node-RED, create a visual display for the log files. The events registered in `nfc_sensor.log`, `doorbell.log` and `reader.log` should be displayed in a ui-table, using the image below as a guide.

NFC Log			Doorbell Log			Reader Log		
Time	Card Type	UID	Date	Time	Status	Time	Action	Status
2023/06/21 16:08:40	MIFARE Classic 1K	4BF2ACF8	2023/06/21	16:08:41	Failed	2023/06/21 16:09:31	Buzzer	Success
2023/06/21 16:08:45	MIFARE DESFire	042149423D4680	2023/06/21	16:08:48	Failed	2023/06/21 16:09:38	Buzzer	Success
2023/06/21 16:08:50	MIFARE DESFire	042149423D4680	2023/06/21	16:08:57	Success	2023/06/21 16:10:03	Orange LED Blink	Success
2023/06/21 16:09:15	MIFARE Classic 1K	4BF2ACF8	2023/06/21	16:09:20	Success	2023/06/21 16:10:13	Orange LED Blink	Success
2023/06/21 16:09:30	MIFARE Ultralight C	04EE5E4A934F80				2023/06/21 16:10:20	Buzzer	Success
2023/06/21 16:09:37	MIFARE Ultralight C	04479982F05C80				2023/06/21 16:10:29	Buzzer	Success
2023/06/21 16:10:00	MIFARE Classic 1K	4BF2ACF8				2023/06/21 16:11:33	Buzzer	Failed
2023/06/21 16:10:11	MIFARE DESFire	042149423D4680				2023/06/21 16:11:38	Orange LED Blink	Failed
2023/06/21 16:10:20	Unknown	058BA901EAA100				2023/06/21 16:12:32	Buzzer	Failed
2023/06/21 16:10:29	MIFARE Classic 1K	0477D2BA153C80				2023/06/21 16:12:48	Buzzer	Failed
2023/06/21 16:11:33	MIFARE Classic 1K	6E924C40				2023/06/21 16:13:01	Buzzer	Failed
2023/06/21 16:11:38	MIFARE DESFire	042149423D4680						
2023/06/21 16:12:32	MIFARE Ultralight C	04EE5E4A934F80						
2023/06/21 16:12:48	Unknown	058BA901EAA100						

Use `paho.mqtt` to publish messages, e.g. UIDs, Doorbell ring events etc, and Node-RED *mqtt in* nodes to listen for these messages. The messages would need to be formatted as seen in `mqtt_publish.py` on Moodle. Doorbell failed events can be triggered by disconnecting the `hackrf` from

the computer and the Reader failed events can be triggered by referencing the incorrect reader index (e.g. `r[0].createConnection()` VS `r[1].createConnection()`).

The table columns:

- For `nfc_sensor.log`:
 - Time – the date and time the card was detected
 - Card Type – the NFC card type e.g. Ultralight C
 - UID – the UID of the card
- For `doorbell.log`:
 - Date – the date the doorbell rang
 - Time – the time the doorbell rang
 - Status – if the doorbell rang successfully or failed
- For `reader.log`:
 - Time – the Date and Time the event occurred
 - Action – orange LED or buzzer event
 - Status – if the action was successful or failed

The Node-RED code should be exported and saved as a json file for submission.

Due Date: Check Moodle for date

Late Submission Deadline: Check Moodle for date (usually 1 week after due date)

Submission Files: python scripts

- `processor.py`, `acr_sensor.py`, `acr_actuator.py` and `hackrf_actuator.py`
- The GNU python script that was used to configure the doorbell, *if used*
- The doorbell data files should not be uploaded
- The Node-RED `json` file if the Bonus is attempted

Location: Moodle Homework12 link