Alexandru - Cristian Dumitrache
adumitrache@tudelft.nl

# Voltage Supply for Liquid Crystal Reconfigurable Intelligent Surface Biasing

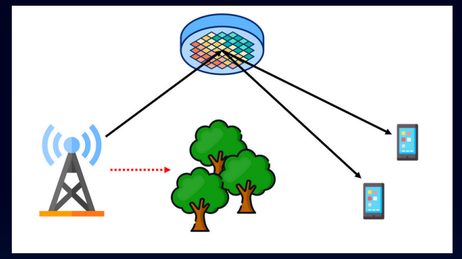
Supervisors:
Arash Asadi & Fabian Portner

## 1. Introduction

### RIS (Reconfigurable Intelligent Surface)

An RIS is a meta-surface composed of an arbitrary number of radiating elements, which can be individually tuned to manipulate EM (electromagnetic) waves.

An LC-RIS is a type of RIS that uses liquid crystal materials.



## 2. Research Objective

### Research Question:

"What is a scalable, low-latency hardware-software solution for controlling high-speed LC-RIS systems?"

### LC-RIS Reference System:

"Architecture for sub-100 ms liquid crystal reconfigurable intelligent surface based on defected delay lines" by Neuder et al.

### Technical Requirements:

- Independent voltage control for ≥ 50 output channels.
- Software controllable, preferably with Python support.
- Output waveform: Square wave at 1 kHz.
- Voltage range: Minimum ±10 V.
- Resolution: At least 7-bit.
- Fast response: Voltage updates within < 1 ms.

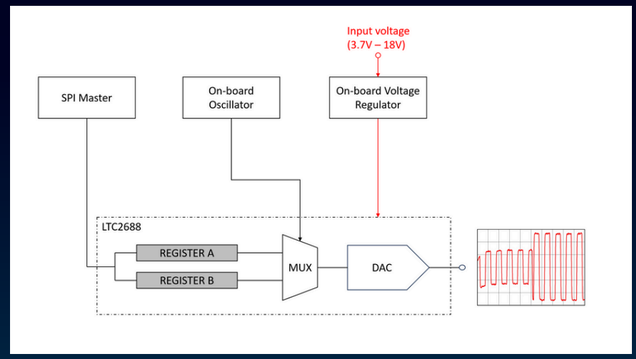
## 3. System Design

### Used setup: DAC60096EVM

- Slow response times - update of 64 channels - 161 µs
- Deprecated

### New setup: 4 x LTC2688

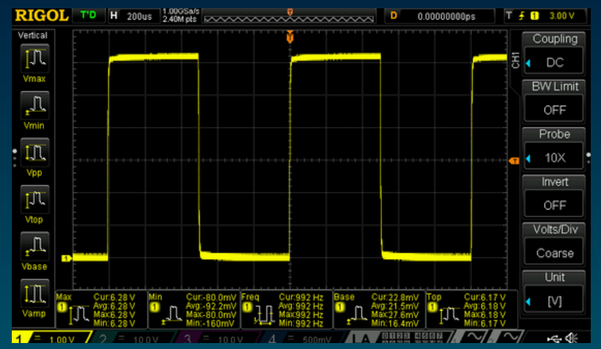
- Faster response times - update of 64 channels - 62 µs
- Daisy - chain configuration
- In-built power supply
- In-built oscillator for 1KHz square wave generation
- Scalable and flexible




## 4. Results

### Output Analysis

- Configured at a voltage range of ±15V
- Code 2890; expected 6.17 V, measured at 6.22 V
- Expected 1Khz square waveform, measured at 992 Hz
- Perfect duty-cycle of 50%




### Response Time Analysis

- All tests - SPI frequency of 25 MHz
- Best Time - Update of one single channel
- Worst Time - Update of all 64 channels
- Two code style implementations:
  - Minimalistic
  - Object Oriented (OO)
- Best result: Raspberry Pi Pico in minimalist C

| Device | Best Time µs | Worst Time µs |
|---|---|---|
| Ideal | 26 | 103 |
| Raspberry Pi 5 (minimalist C) | 50 | 813 |
| Raspberry Pi 5 (minimalist Python) | 60 | 974 |
| Raspberry Pi 5 (OO Python) | 99 | 1600 |
| Raspberry Pi Pico (minimalist C) | 40 | 273 |
| Raspberry Pi Pico (minimalist MicroPython) | 57 | 935 |
| Raspberry Pi Pico (OO MicroPython) | 3680 | 82120 |
| PC + FT232H (minimalist Python) | 1430 | 24050 |
| PC + FT232H (OO Python) | 1780 | 24250 |

## 5. Conclusions

### Future Improvements

- SPI master needs further analysis and development
- Frequency can be improved by adding manual tuning

### Overview

- System achieves desired output, fulfilling requirements
- Can adapt for changes in the LC-RIS requirements