SafeVision

Nathan Bornfreund, Rick Franc, Steven Landis, Willy Tu

The Problem

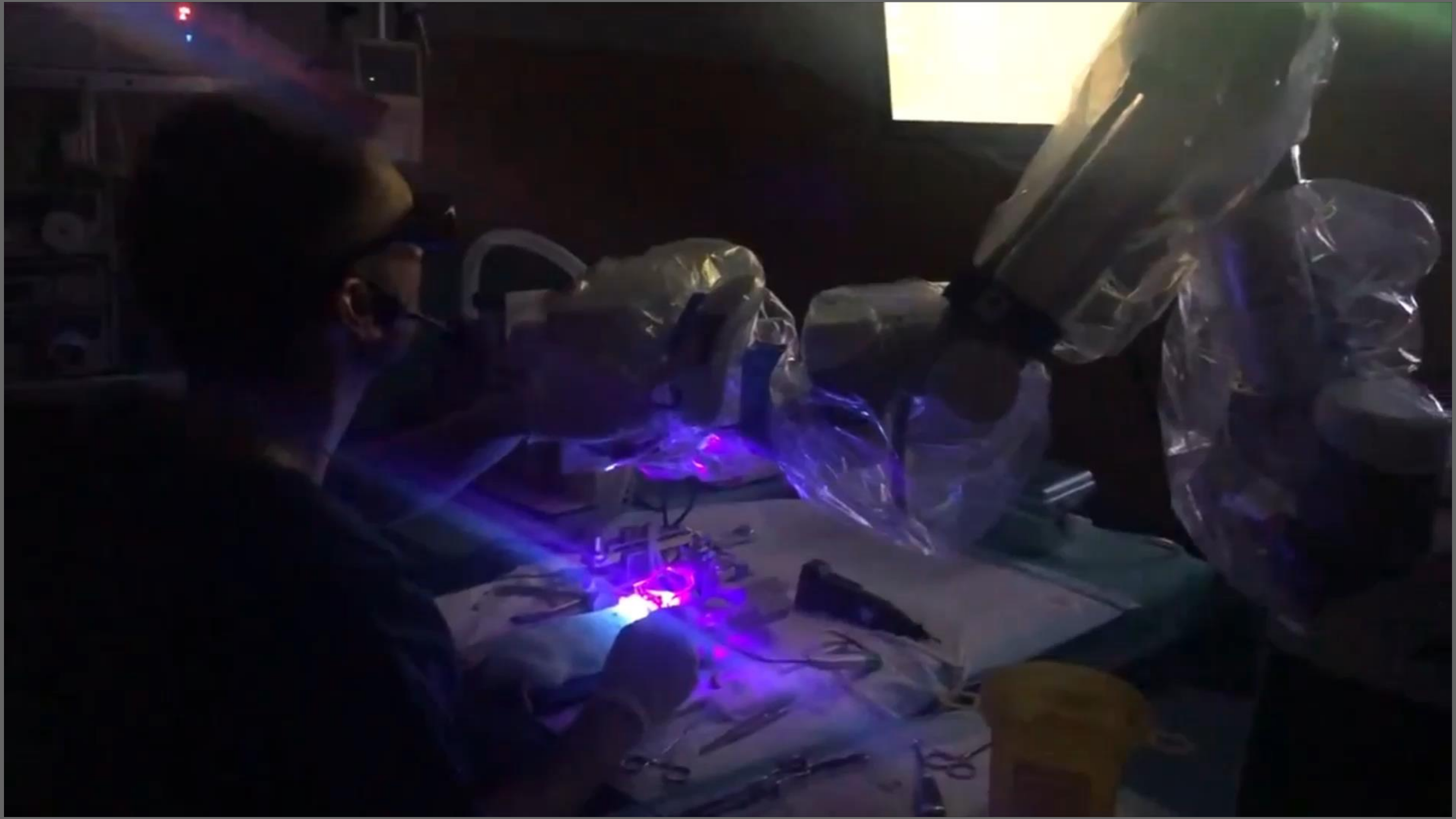
SafeVision

- Robotic arm with camera attached used for surgeries.
- The arm can collide with patients, doctors and equipment.
- The arm will stop, but only after running into something.
- If only there was some way to stop *before* hitting something...

Applications

SafeVision

- Assisted Drive
- Free Drive
- Rest Mode
- Remotely Controlled



Requirements

- Prevent object collision
- No unexpected movements
- No purely autonomous motion
- All parts have to be approved for medical environments

The Solution

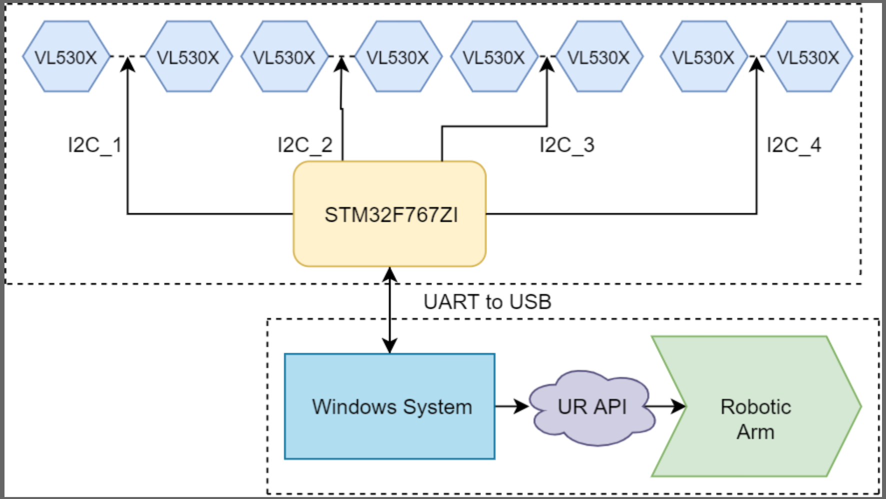
SafeVision

- Create spatial awareness using a matrix of distance sensors
- Check sensors in direction of movement
- Stop movement if an object is detected

Project Description

SafeVision

- Microcontroller
  - Optical Distance Sensor
    - I2C
  - Send distance data to the main system through UART
- Main Window System
  - Process distance
  - Control the robotic arm.



System Expectations

SafeVision

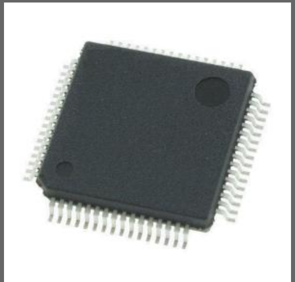
- Detect people and objects before colliding.
- Take measurements at least 2Hz.
- PCB must be small enough to fit on the joint of the arm.

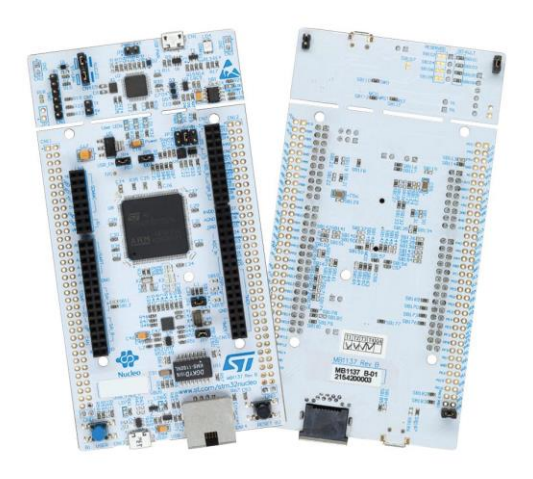
Processor

SafeVision

STM32F767ZI

- High-performance
- DSP with FPU
- M7 processor with 216 MHz clock and floating point instructions
- 4 - I2C busses
- UART communication
- ST Link Debugger

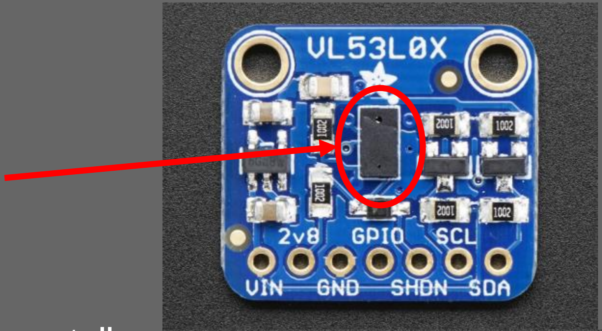


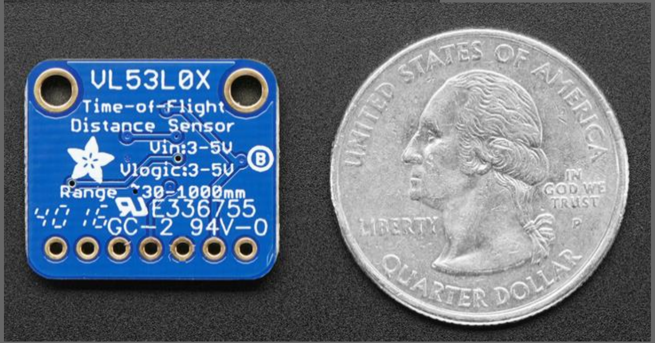


Sensors

VL53L0X Time of Flight Sensor

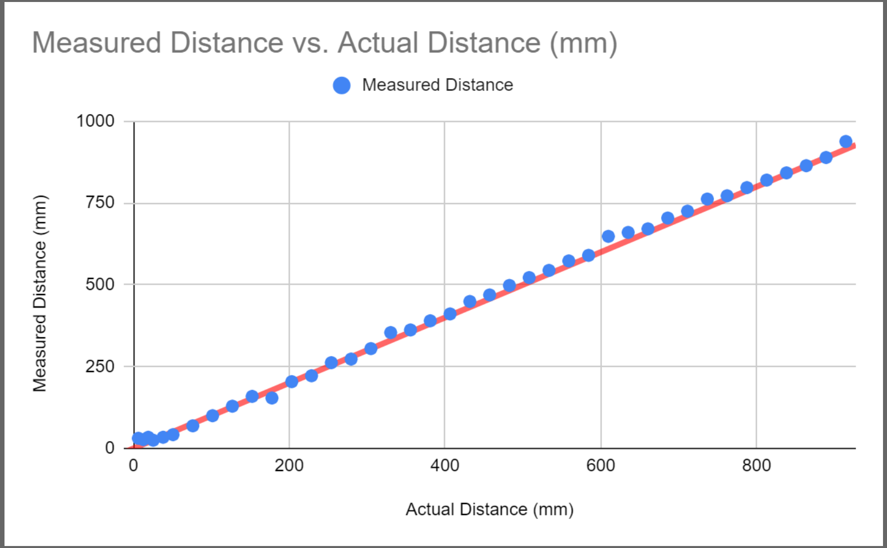
- Takes 20ms to get measurement.
- Accurate from 2.5 cm to 80 cm (experimentally verified).
- Communicates over I2C.

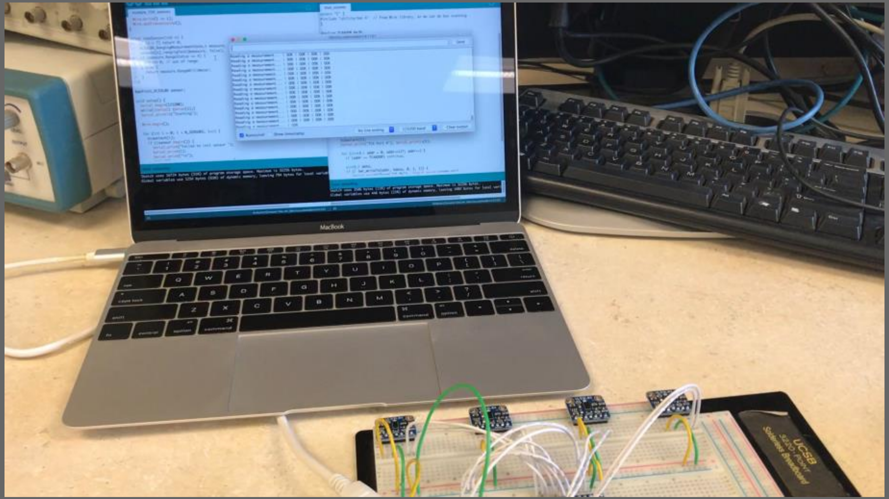




Testing

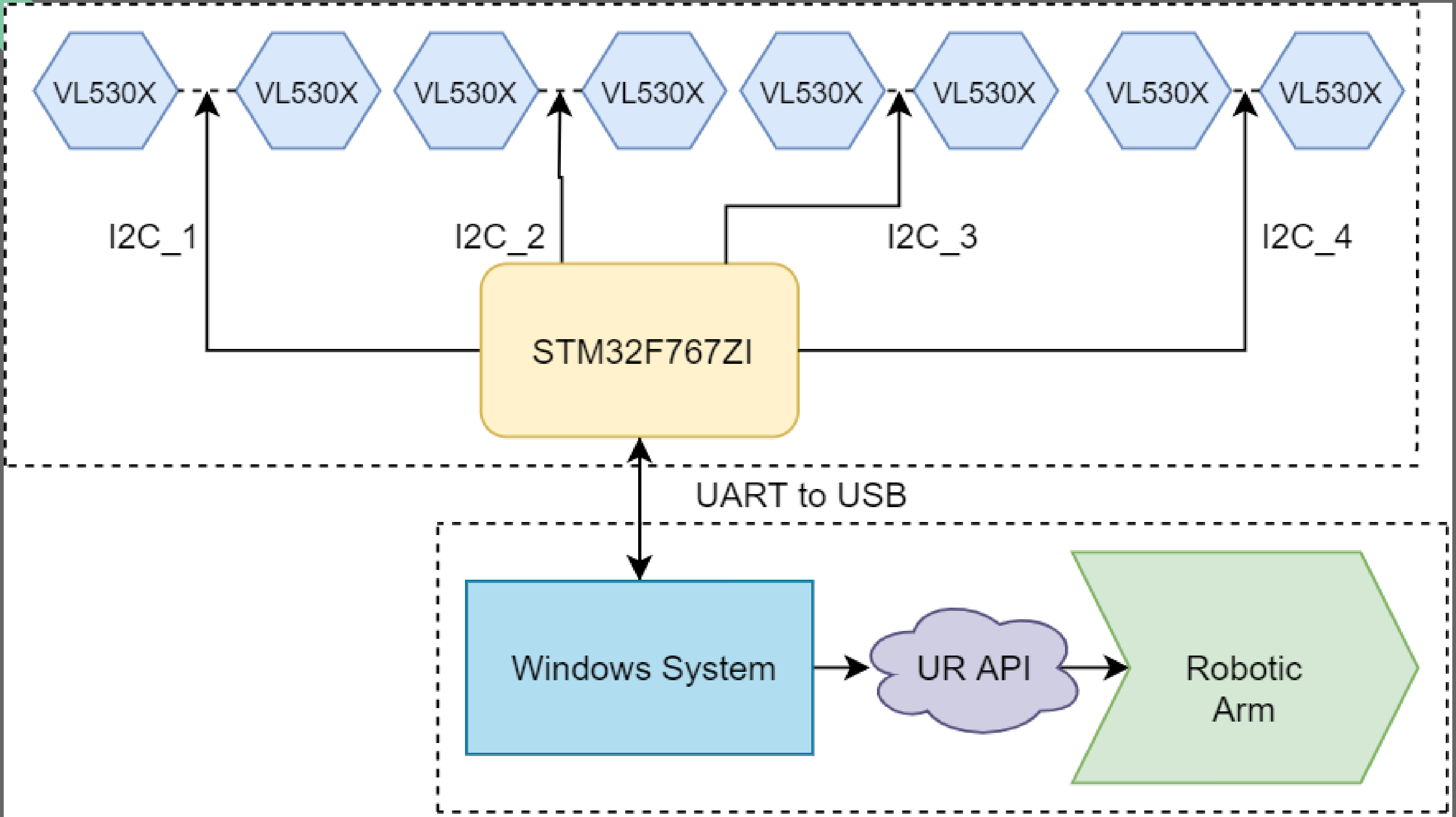
SafeVision





Overall Project Structure

SafeVision

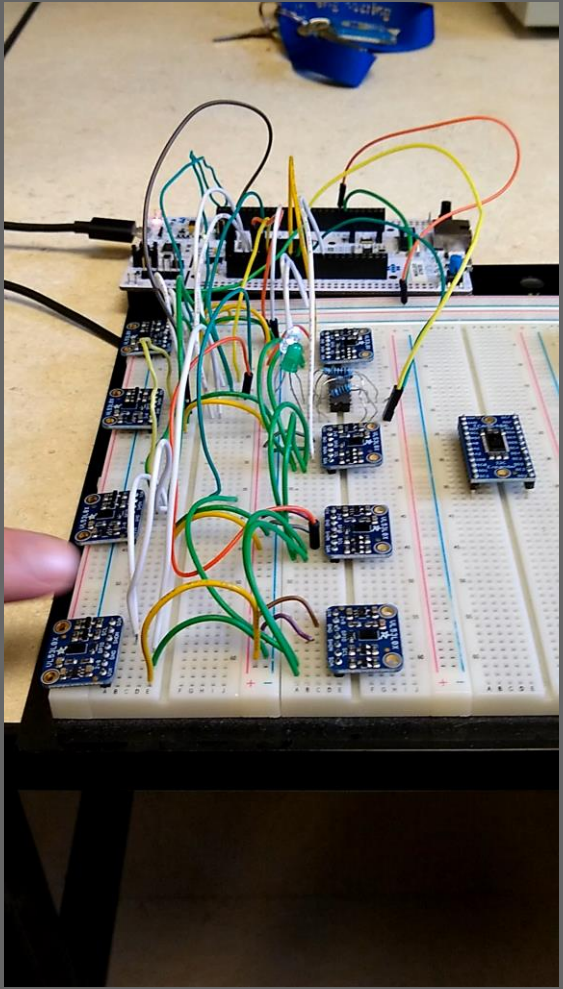


Microcontroller Software Structure

- Init Peripherals
- Setup VL53L0X Sensors and update device addresses
- Start OS Scheduler with 4 thread using RTOS
- In each thread,
    - Responsible for different I2C bus
    - Collect distance data from all device on the bus
    - Send distance data to the main window system through UART.
        - Each data point is labeled with a UID
- Mainly focus on optimizing the code to decrease the total read time.

More Testing

SafeVision



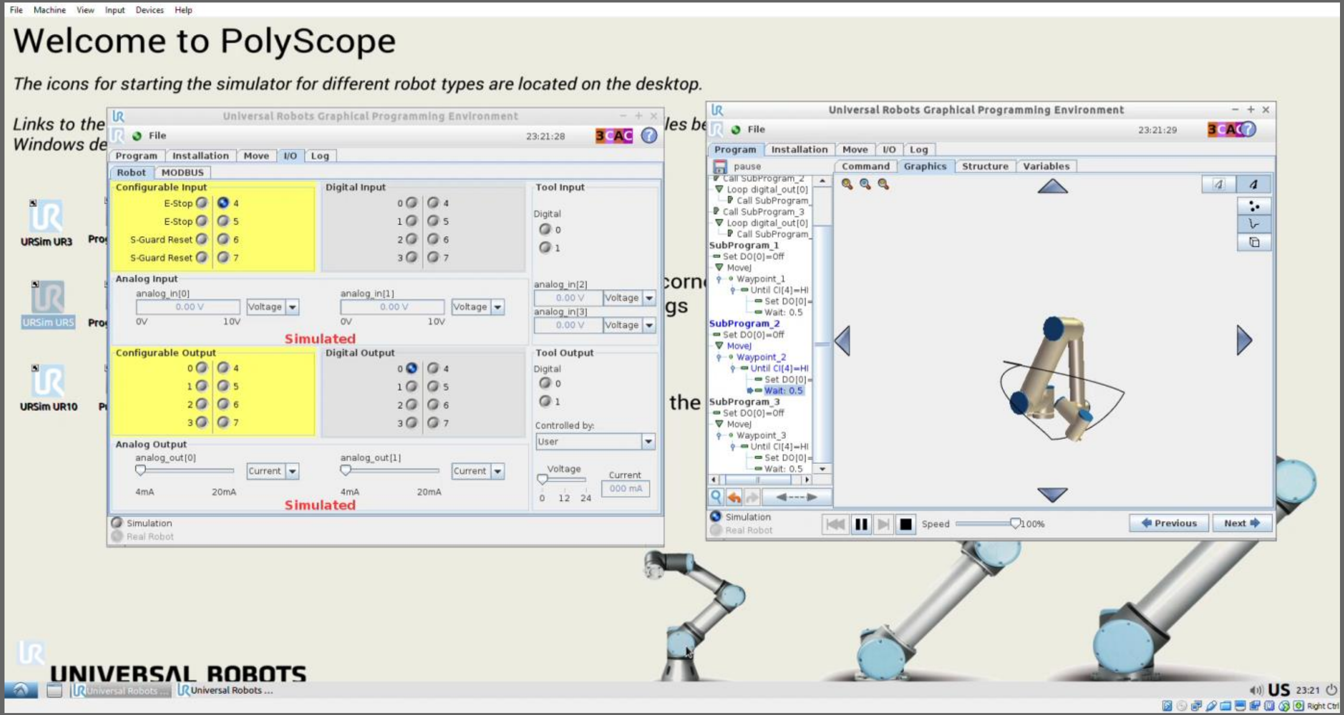
Windows System Software Structure

SafeVision

- Process data received from the microcontroller
- Main System ←→ Universal Robot Arm
  - Communicate through API calls
  - Receives information of the arm’s position and movement direction
- Use data from microcontroller and arm to determine if an object is in the path of motion.

UR Simulator

SafeVision



Welcome to PolyScope

The icons for starting the simulator for different robot types are located on the desktop.

Links to the ... Windows de...

Summary

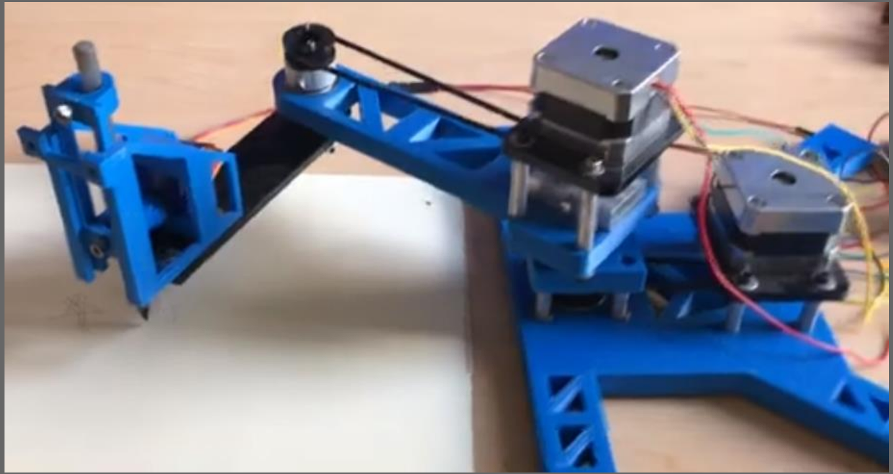
SafeVision

- Experimented with different sensors and decided that VL53L0X will be satisfactory for our project
- Tested on a Arduino and STM32 development board
- Tested software functionality of the processor to match project needs
- Picked out the processor and started designing the PCB
- Tested and decided on communications between interfaces
- Set up a test movement program in UR simulator to be stopped through an input

Goals

SafeVision

- Have complete PCB built by early winter quarter
- Set up all 30 sensors on a robotic arm as a test
- Have responses turn off movement of the arm and in UR simulation
- Prepare to test our project on the real UR arm and camera head



Acknowledgements

SafeVision

Yoga, Adi, and Kyle

Yuepei, Sung In, Pat, and Vincent from Alcon

Questions?

SafeVision