**Real Time Systems Design Final Project**

**Course:** ENGG\*4420 Real-Time System Design

**Instructor:** Radu Muresan

**Student Names/Numbers:**

Emeka Madubuike (0948959)

Aaron Hart (0969188)

Mohammed Al-Fakhri (0982745)

Andreas Koumoundouros (0975415)

**Date:** 29/11/2018

Contents￼

Contents

[1 Introduction 3](#_Toc25916970)

[1.1 Problem Description 3](#_Toc25916971)

[1.2 System requirements 3](#_Toc25916972)

[2 Background 3](#_Toc25916973)

[2.1 Short overview of the advantages of using LabView 3](#_Toc25916974)

[2.2 STM32F429 3](#_Toc25916975)

[2.3 PIR Motion Sensor 4](#_Toc25916976)

[2.4 Arducam 4](#_Toc25916977)

[3 Implementation 5](#_Toc25916978)

[3.1 Block Diagram 5](#_Toc25916979)

[3.2 Implementation details 5](#_Toc25916980)

[3.3 Simulation results 5](#_Toc25916981)

[4 Sources of Error. 7](#_Toc25916982)

[5 Future Steps 7](#_Toc25916983)

# 1 Introduction

## 1.1 Problem Description

In this project, students were asked to develop a more complex real time application. As a project, students are required to build a home security system. Students can use either FreeRTOS or uC/OS-III. The system needs to detect the intruder, when the system detects an intruder the system needs to capture the image of the intruder and display a live feed on the STM32F429. The project is required to have 3 tasks with different priorities, and utilize RTOS objects for task synchronization, communication etc.

## 1.2 System requirements

The system was to be made using a PIR motion sensor to detect any intruders, a basic arducam to capture images, and a STM32F429 Discovery board to interface and control everything. The program for the project needs to activate an interrupt whenever the motion sensor is triggered and needs to send a message to the main board whenever this happens; when this happens the board will display a message that there is an intruder, and will take a picture with the arducam to record the intrusion. As an additional feature, the camera can also capture an image whenever the user manually triggers it with the user button on the discovery board.

# 2 Background

## 2.1 Short overview of the advantages of using LabView

The primary goals of this laboratory were to learn both uC/OS-III and FreeRTOS operating systems as well as implement a simple producer-consumer task scenario. However, LabView was used to show how such a system could be implemented within our plant design from lab 1. By using a queue system for data input and output within the lab 1 model we were able to determine that a synchronized task system would be essential for implementing our plant model in the uC/OS-III or FreeRTOS operating systems.

## 2.2 STM32F429

This apparatus allows for the making of interactive applications. The STM32F429 lines offer the performance of the Cortex-M4 core running at 180 MHz. At this frequency the STM32F429 delivers high performance executing from Flash memory with zero wait times due to ST’s ART Accelerator.  The DSP instructions and the floating point unit enlarge the range of addressable applications. ST’s 90 nm process, ART Accelerator and the dynamic power scaling enables the current consumption in run mode and executing from Flash memory to be as low as 260 µA/MHz at 180 MHz.

## 2.3 PIR Motion Sensor



The Pyroelectric InfraRed Sensor is a light sensor that detects the presence of infrared light emitted from a warm body, or an interruption in a constant infrared light that is emitted. This PIR module has 3 pin connection consisting of power supply, ground and an output signal that turns logic high when movement is detected. The sensor uses 3 to 5 V input.

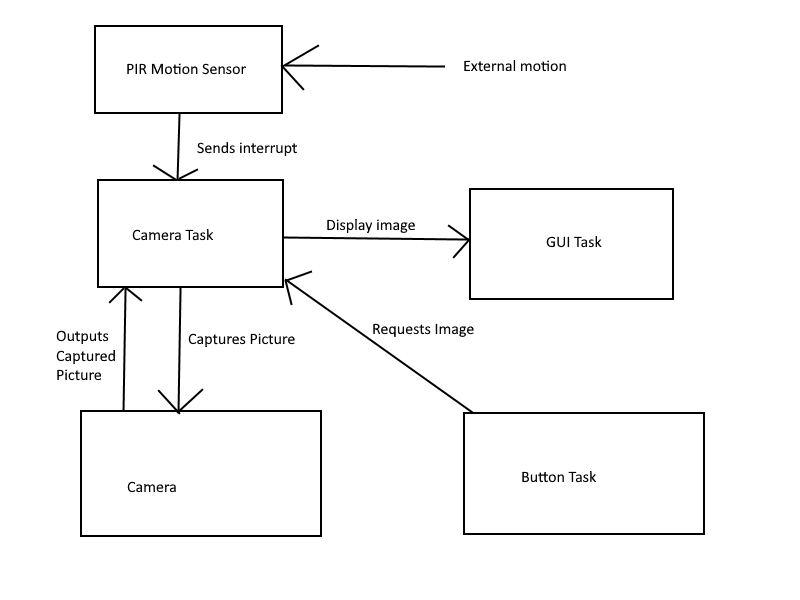
## 2.4 Arducam



ArduCAM is a universal camera control board for Arduino. It hides all the complexity nature of a camera. This camera provides different format like RAW, RGB, YUV, JPEG. This camera is a 5MP camera. It uses I2C for camera control, SPI for ARDUCHIP, and lastly it can be connected to the STM32 board via the I2C3 and SPI 4.

# 3 Implementation

## 3.1 Block Diagram



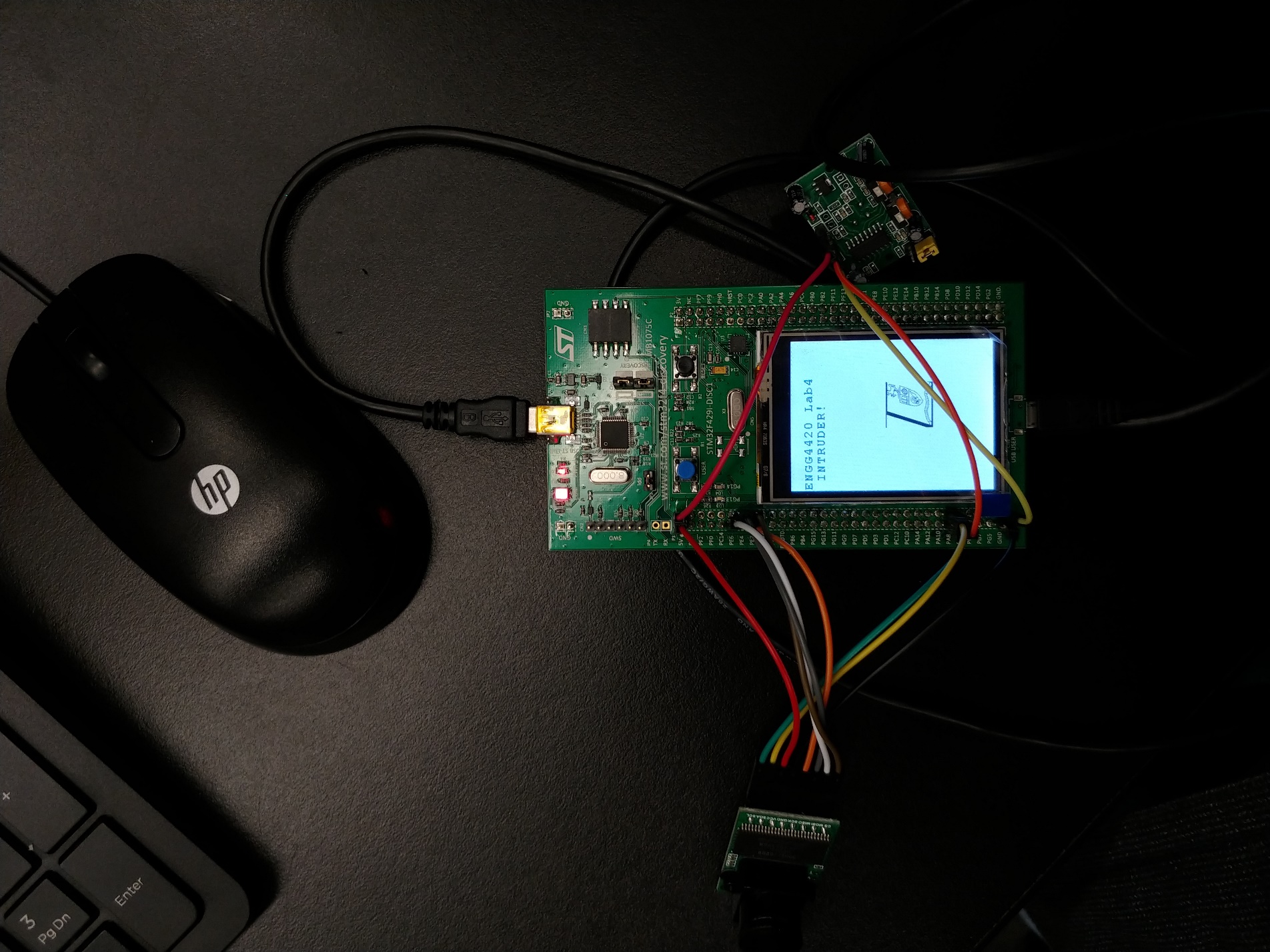
## 3.2 Implementation details

To implement this program, the project was written with FreeTOS. 3 tasks are used to run the program: one for capturing images, one for updating the GUI, and one for listening for the user button to be pressed. When the motion sensor is triggered, an interrupt is enabled, which sets a global flag in the code. The task for recording with the camera waits for the said flag to be set, and when it is done the task resets the flag, captures an image, and gives an output to the main board. The Button task works similarly, except it waits for the user button to be pressed rather than waiting for an interrupt.

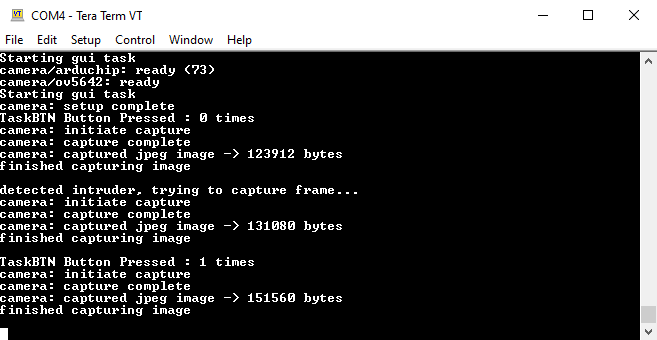
The motion sensor was interfaced to the board through pin PC8. The Camera is connected through pins PA9, PA8, PE2, PE3, PE5 and PE6 for the inputs SDA, SCL, SCK, CS, MISO and MOSI respectively.

## 3.3 Simulation results

Below is an image of the GUI of the discovery board after an intruder has been detected. A sample image is displayed below where it informs the user that an intruder has entered.



Below is an image of the console output after a couple of intruder detections and a couple manual camera triggers.



# 4 Sources of Error

The image captured by the arducam is recorded but unable to be displayed on the LCD display. This was a bug we encountered during development because the line of code in the GUITask which cleared Str4Display each iteration was commented. This caused some form of interference with the function which draws bitmaps to the LCD resulting in an inability to display images while using the camera.

# 5 Future Steps

Due to time constraints we were unable to save the transferred image to the board and convert that transferred image to a format that can be displayed on the LCD. Going forward we would implement a data structure within the code to save the image as it is read from the arducam board. Using this saved image, we could then do the necessary conversion from jpeg to bitmap to make use of the BSP\_LCD\_DrawBitmap function.

Based on the compressed jpeg size as reported in the print statements we would likely require some form of downscaling in order to convert the image into a usable format that could be display on the board. Additionally, initial camera setup parameters could be set so that the captured image is of a lower resolution such as 320x240 to allow for faster conversion, transfer, and processing. At this resolution we could conceivably show a continuous low-framerate video feed on the LCD that can freeze on the current frame when an intruder is detected. The performance of this possible functionality would be limited only by the main board’s processing capabilities.

6 References

Data acquisition. (2019, April 16). Retrieved from <https://en.wikipedia.org/wiki/Data_acquisition>.

ENGG4420: Real-Time Systems Design Lab Manual. (2019, August). Retrieved from <https://courselink.uoguelph.ca/d2l/le/content/577069/viewContent/2190218/View>