Date: March 19th, 2019

From: Adam Kwok

To: Doctor Kaputa and anyone else interested in my work

Subject: PyQT4 utilizing OpenCV Tennis Ball Detection Analysis

Introduction

For the CPET-563 drone project, we utilized OpenCV for the detection and tracking of tennis balls. The important factors in this design are that it was developed to include capabilities of loading and saving configurations from a file,

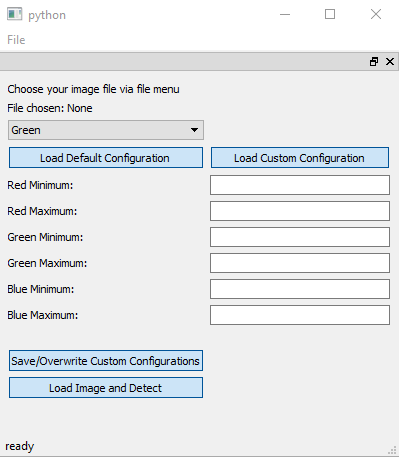
For the development of a PWM designed in VHDL for future use in running motors for a drone, Modelsim was utilized for simulation/verification. The important factors in this design were that the PWM module required an enable pulse, control of both the duty cycle and period, and implementation with only basic entity inputs, eliminating the use of generics for period and duty cycle inputs. The reason for these requirements were to ensure the following:

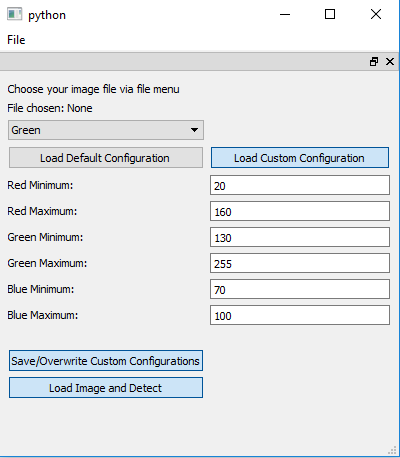
* Establish a strong foundation for future modules and modifications to build upon
* Verify functionality in the environment with a simple, yet important base
* Determine limitations of PWM inputs
* Determine error inputs for future modules

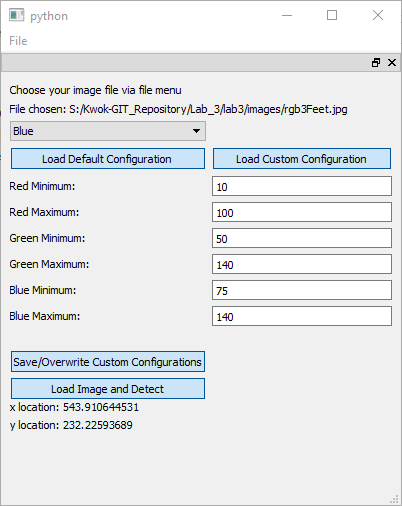
The below analysis investigates the limitations and errors in different cases established and recognized for the future implementations. This module was designed for scalable usage with different devices, but the intended use of being implemented on a Snickerdoodle’s FPGA. This memo investigates not only the limitations of the FPGA implementation, but the recommended additions to be added to end-user GUI interactions to dictate the limitations of user input on the system.

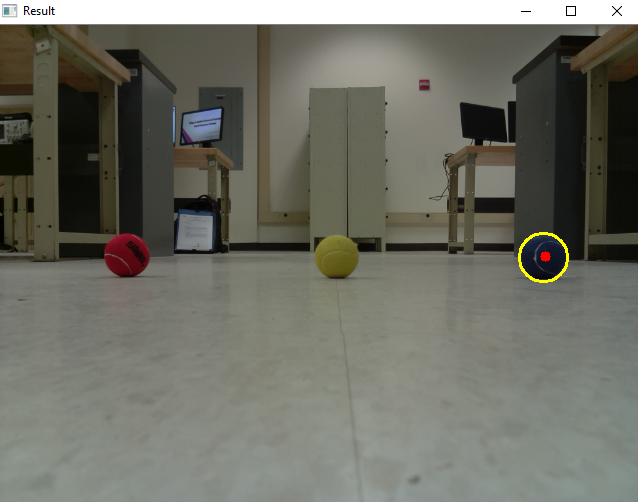
Analysis

With the goal of making the only fault possible occur outside of this module, the module was designed such that all errors were addressed. The inputs included the clock, reset, enable, period, and duty cycle while the outputs included an error signal and the pwm pulse output. The reset set a count signal, the pwm pulse, and the error signal all to zero.









Conclusion

* Because this code is set to interact with a 100 MHz Snickerdoodle FPGA clock, the user should be informed of recommend minimum and maximum periods and duty cycles based on the associated motors
* Because the error signal is held high, the developer should create a graphical alert so that the error can be recognized in order to be avoided in the future
* Since this project is designed to interface with a 100 Mhz Snickerdoodle FPGA clock, the code that interacts with this module should be well-commented with the information necessary to results of this module being accurate based on device specifications
* The user should be informed of the frequency, period, and duty cycle at all times in an interface