Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Summer 2023

Ganbare
Automated Wii Play Shooting

Alexander Olmedo
Duy Nguyen
Jack Gervasi
Justin Geisen
Lam Do

Revision History

Revision	Date	Author(s)	Description
0.1	7.07.2023	JuG	document creation
0.2	7.14.2023	AO, DN, JaG, JuG, LD	complete draft
1.0	8.08.2023	AO, DN, JaG, JuG, LD	official release

CONTENTS

1 Introduction **5**

2 Subsystem Definitions & Data Flow **6**

3 Wii U Layer Subsystems **7**
3.1 Layer Hardware . . . 7
3.2 Layer Operating System . . . 7
3.3 Layer Software Dependencies . . . 7
3.4 Console . . . 7
3.5 IR Sensor . . . 7
3.6 Wii Controller (held by robot) . . . 8

4 Robot Layer Subsystems **9**
4.1 Layer Hardware . . . 9
4.2 Layer Operating System . . . 9
4.3 Layer Software Dependencies . . . 9
4.4 Robot Arm . . . 9
4.5 PLC . . . 10
4.6 Power Supply . . . 10

5 Control Layer Subsystems **11**
5.1 Layer Hardware . . . 11
5.2 Layer Operating System . . . 11
5.3 Layer Software Dependencies . . . 11
5.4 Capture Card . . . 11
5.5 Image Processing . . . 11
5.6 Robot Control . . . 12

6 Appendix A **13**

List of Figures

1 A simple data flow diagram 6
2 Wii U layer subsystems 7
3 Robot layer subsystems 9
4 Control layer subsystems 11

1 INTRODUCTION

The goal of this project is to get high schoolers interested in UTA computer engineering. This project is meant to be very interesting and fun to watch. A robot arm will play a shooter video game by receiving data from the screen, identifying the target, and pointing and shooting at the target quickly. This project has deviated from previously using a robot arm to play Duck Hunt on the Wii, because the Wii console and online store has been discontinued. This game was only offered through the online Wii store. This is why we are now using Wii Play which is a hard disk game that was purchased from Amazon.

2 Subsystem Definitions & Data Flow

The figure below displays how different components of the system interact with each other in more detail. The Wii U consists of the console, the IR sensor, and the Wii controller held by the robot. The console will provide data to the television and the control layer through an HDMI splitter. The control layer will receive data from the console through a capture card which will relay the data to a computer for image processing. After image processing is done, instructions will be sent to the robot through a PLC. Then, the physical robot arm will move holding the Wii controller.

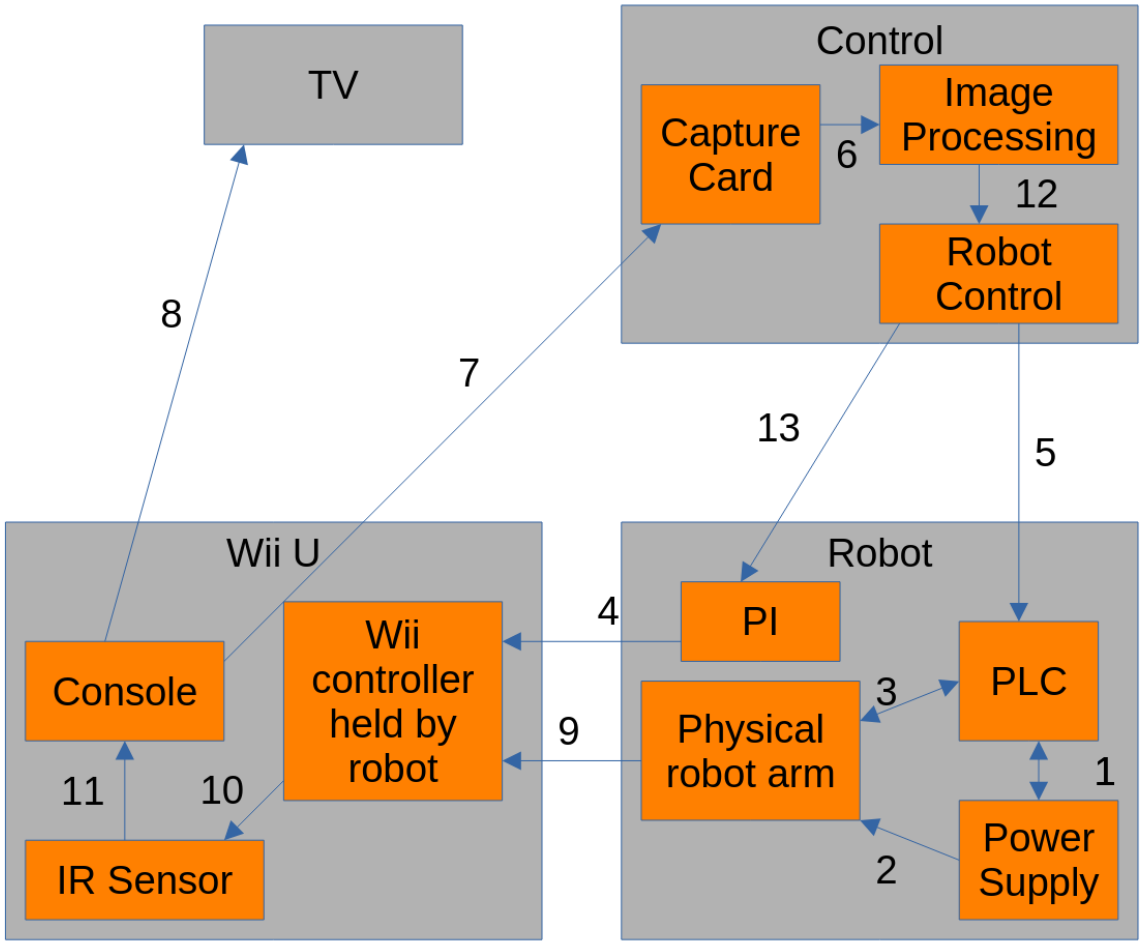

Figure 1: A simple data flow diagram

3 Wii U Layer Subsystems

This layer will be generating data to be displayed on the television and the control system. The control layer and the robot layer will be controlling the Wii remote which will affect the game.

3.1 Layer Hardware

The Wii U layer is the layer of which we are automating. It provides the game that will be played and runs the game.

3.2 Layer Operating System

The Wii U system software is both the official firmware and the operating system for the Wii U.

3.3 Layer Software Dependencies

None

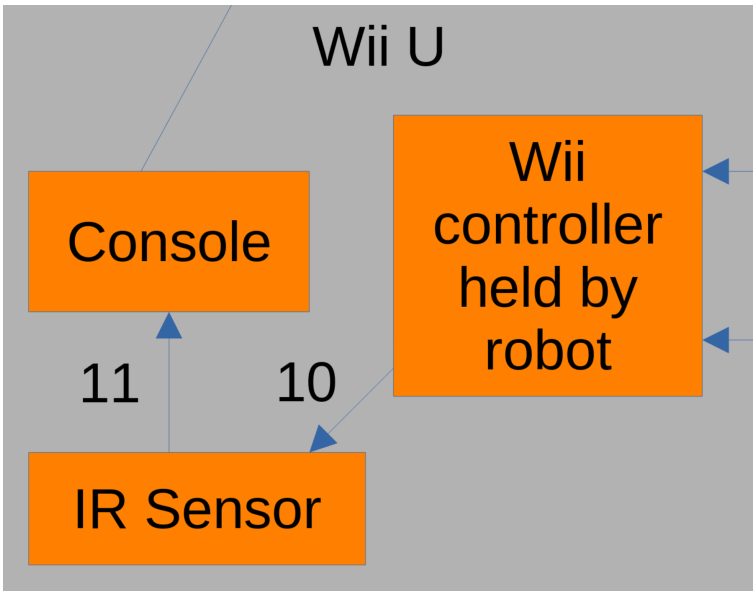

Figure 2: Wii U layer subsystems

3.4 Console

The console provides the game that is being played and contains the hardware needed to run the game. The Wii U console will receive input solely from the Wii controller. The Wii U will display the output into an HDMI splitter which will forward the data to the TV screen and the capture card.

3.4.1 Subsystem Hardware

The console is made of a Wii U console and all its respective components sans the IR sensor.

3.4.2 Subsystem Operating System

The Wii U system software is the official firmware and operating system for the Wii U.

3.5 IR Sensor

The IR sensor detects the Wii remote and allows for cursor movement across the screen from that respective Wii remote.

3.5.1 Subsystem Hardware

The IR sensor is a small rectangular discrete device that plugs directly into the Wii U Console.

3.6 Wii Controller (held by robot)

The Wii controller is what the player uses (or in this case the robot arm) in order to play the game. The Wii controller button will be operated through soldered wires hooked up to a Raspberry Pi. The Raspberry Pi will be giving the commands to the Wii controller's buttons.

3.6.1 Subsystem Hardware

There is a slight change from a typical Wii remote in that we soldered wires that if they are connected physically, it is equivalent to a trigger press. The Wii controller will be inside a 3d printed case. The case will have a mount that can be screwed into the robot arm.

4 Robot Layer Subsystems

This layer consists of the physical robot arm, the programmable logic controller (PLC), and the power supply. This layer is responsible for the movement of the physical robot arm and receives data from the control layer.

4.1 Layer Hardware

There are cable connections between the PLC and robot arm so that they can communicate with each other.

4.2 Layer Operating System

The robot will be using ROS2.

4.3 Layer Software Dependencies

The Python libraries that are being used are rtde_control and rtde_receive.

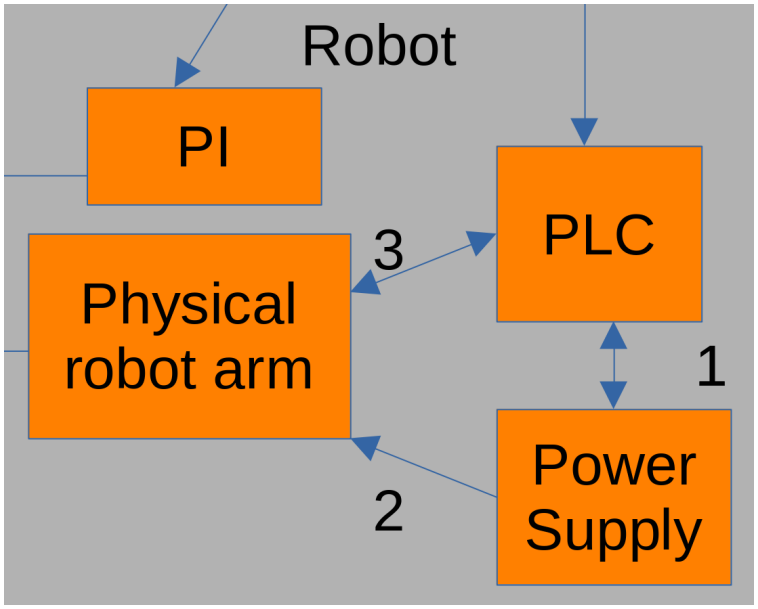

Figure 3: Robot layer subsystems

4.4 Robot Arm

The physical robot arm will have the Wii remote attached to it. The robot arm will receive input from the PLC. It will point the Wii remote at a designated angle.

4.4.1 Subsystem Hardware

The physical robot arm has an attachment area where screws can be placed. The robot arm has 6 axes of rotation. The physical robot arm is made by Universal Robots with model UR5.

4.4.2 Subsystem Data Processing

We used a linear projection algorithm to calculate the angular position of the robot arm. Once the desktop computer knows the coordinates of the target on the screen, this algorithm can be used to give the exact angles for the robot arm to use. Although this algorithm is fairly accurate, the IR sensor is inaccurate at measuring exactly where the Wii remote points. We are currently developing another

algorithm that will adjust the Wii cursor based on the Wii cursor's position relative to target on the screen purely in the virtual space. This will algorithm will adjust for the IR sensor's inaccuracy.

4.4.3 Subsystem Software Dependencies

Python 3.10.6 was used to control the robot arm.

4.5 PLC

The PLC works as an intermediary device between the computer and the robot arm. It gives the commands from the computer to the robot arm.

4.5.1 Subsystem Hardware

The PLC is a metal box that is placed underneath the robot arm.

4.5.2 Subsystem Data Structures

The data is transmitted over Ethernet which goes through 2 switches. This lets the computer control the robot arm while also receiving feedback from it.

4.6 Power Supply

The power supply provides power to both the PLC and the physcial robot arm.

5 Control Layer Subsystems

For the purpose of target identification and acquisition both software and hardware will be implemented. Due to the nature of the available robot arm and its current python language, our programming language will be implemented in python while making use of OpenCV libraries for computer vision.

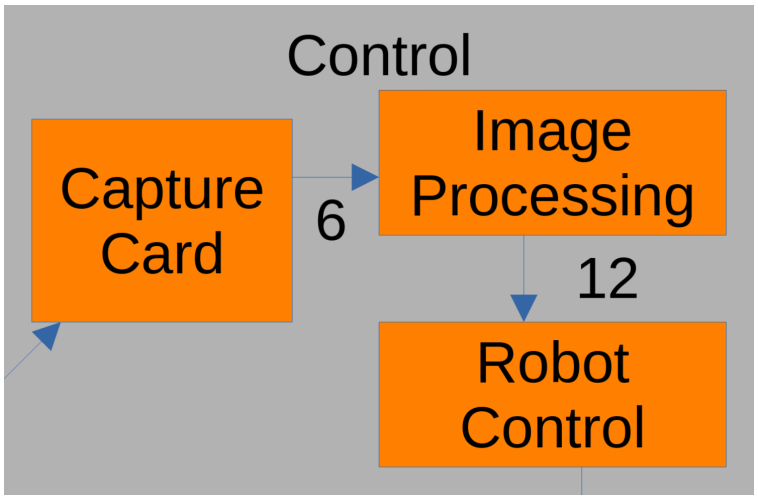

Figure 4: Control layer subsystems

5.1 Layer Hardware

This will include a computer and a capture card. The computer will handle the image processing as well as sending the robot control instructions to the PLC.

5.2 Layer Operating System

The computer will be running Ubuntu 22.04.2 LTS.

5.3 Layer Software Dependencies

The control layer will use NumPy and itertools.

5.4 Capture Card

The capture card is plugged into the computer and is used to capture the display from the Wii U. For image capturing, OpenCV will capture the image in video format, then splice into individual frames for processing from the image capture card. The image will also be mirrored by the capture card, onto a TV for the robot to engage.

5.4.1 Subsystem Hardware

The capture card takes the output signal from either the console or HDMI splitter converting it into an input device that can read into the computer.

5.4.2 Subsystem Software Dependencies

The capture card uses OpenCV 4.8.0 to interface with the video capture card to get recordings from the game console.

5.5 Image Processing

For image processing, the image will be fed from the capture card to a computer in real-time.

5.5.1 Subsystem Software Dependencies

The image processing will use Python 3.11.3. In order to process the image, Python 3 script will be implemented in combination with libraries from OpenCV 4.8.0. OpenCV will open the video given the camera path using its library.

5.5.2 Subsystem Data Structures

The video will be spliced at a maximum allowed frames per second and fed through a processing algorithm.

5.5.3 Subsystem Data Processing

For image recognition, the image will be converted to grayscale, then blurred, then normalized. This will make contour detection easier. The background will be captured for a certain amount of frames, for comparison and used in template matching. Template matching will be used in its form of background subtracting to single out regions of interest for target comparison. Once a contour of interest is detected, it will be compared with the targets of interest. The program will output the location of its best-estimate target, and will signal the robot (via ros).

5.6 Robot Control

Once the Image Processing subsystem receives both the coordinate location of the intended target, and its engagement signal, the robot arm will traverse to given location, and signal the engagement button on the Wii U controller. It will continue this identification-engagement loop until it receives a signal that the game event is over.

5.6.1 Subsystem Software Dependencies

The robot control will use Python 3.11.3. The robot control subsystem will use ROS2 libraries in combination with the image processing subsystem.

6 Appendix A

References