Gesture Controlled Claw Machine Conceptual Design

Hou, Chuyi* Li, En Xu* Shentu, Chengnan* Chen, Ryan*

Robotics Institute, University of Toronto

April 9. 2020

*indicates equal contribution

Contents

1 Introduction

Claw Crane is a type of arcade game known as merchandiser. Its popularity has been widely spread since mid 90s [1]. This game is very goal oriented, to claim the prize you want simply by catching the prize. The prize can be of many kinds, stuffed dolls or candies, etc. The player will need to control the claw inside the machine using a joystick on the outside to catch any prize that the player wants. If the player is skilled and lucky, the prize can be claimed.

This conceptual design report will focus on revising the traditional claw machine. A major change on the catch part will be proposed.

2 Update the Old Fashion Way!

"A good tool is an invisible tool. By invisible, we mean that the tool does not intrude on your consciousness; you focus on the task, not the tool." — Mark Weiser , 1993

The original claw machine is using a joystick as the position control of the claw and a push button as the activation of the claw. As the result of today's rapid growing entertaining technologies, we, as a team, think that the claw machine needs an update. A more interactive and more immersive play style should be implemented. Therefore, it leads to the creation of a gesture-controlled claw machine designed in this project.

3 Solution Overview

The solution to revitalize the claw machine is to replace the joystick control input with a wearable device taking gesture input signal to actuate the claw. Research has shown that the demand for gesture-controlled and wireless machines is getting higher and many of these machines are made by using an accelerometer controlled by Arduino boards[2]. As a result, to provide a more immersive and interactive experience to the traditional claw machine, the gesture control component is to be implemented through a wearable device. The wearable device is in the form of a hand strap implanted with a XYZ accelerometer that can measure and record the relative position and acceleration of the user's hand in 3-D space. Based on the change in position of the user's hand, the device can detect a motion being performed and rotate the corresponding motor to match the claw motion with the input motion. The following diagram illustrates the proposed system.

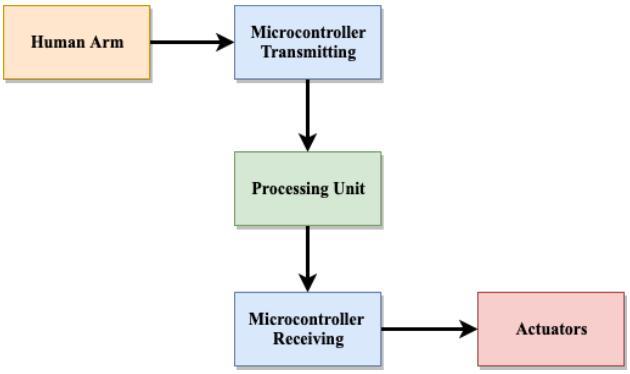

Figure 1: System Overview

4 Summary of Changes from the Proposal

The following list corresponds to a summary of any changes made to the initial proposal of the machine. Details of the change as well as the reason for the change will be discussed thoroughly in each of the designated section.

- The main microcontroller equipped on the machine to control the stepper motors will be Arduino MEGA2560 instead of Arduino UNO. (Section 5.1)
- Three sets of encoders are added to the design to implement a closed-loop feedback control mechanism. (Section 6)
- A state machine will be used to control the claw. (Section 7)
- We will be using NEC protocol with IR to communicate information from the gesture controller to the machine. (Section 9.3)
- The design of the mechanical components are refined for better usability.

5 Microcontroller

The machine will be using 2 microcontrollers, Arduino MEGA2560 and NANO. The purpose of having 2 microcontrollers is to have the machine wirelessly controlled when the user is in the range where RF can be transmitted and received. The MEGA2560 is responsible for receiving RF signals from the gesture-control glove and sending commands to actuate 3 stepper motors to allow the claw to move in 3 degrees of freedom (x,y,z axis). This has been changed from the Arduino UNO microcontroller initially proposed in the project proposal. Due to the extensive usage of interrupts in the software, a microcontroller with more native interrupt pins is the more desirable choice for the project. Since the UNO consists of only 2 interrupt pins and the MEGA2560 allows access to 6 digital pins that could be utilized for interrupts, the team has decided to replaced the UNO with the MEGA2560 instead[3]. In addition, the NANO will be responsible for interpreting the gyroscope outputs due to user hand gestures on the glove and send appropriate signals through the RF transmitter to the MEGA.

5.1 Arduino MEGA2560

The Arduino MEGA2560 will serve as the main microcontroller for the machine and be situated at the top of the claw crane machine. Three A4988 stepper driver boards are used to run the stepper motors. Each board requires 3 digital output pins from the MEGA. In particular, the 3 signals are enable, direction, and step. In addition, the step signal requires a pulse width modulation (PWM) output from the microcontroller. The claw will be actuated through a DC motor, which is controlled by 2 digital output pins from the MEGA. Lastly, a RF receiver is equipped to acquire the user gesture information from the Arduino Nano board (details covered in section 5.2). The RF receiver only needs one input pin. In short, there are a total of 12 digital pins required including 3 PWM outputs. Arduino MEGA2560 is capable of satisfying the needs with a compact design. A full diagram of the Arduino MEGA2560 pin assignment is attached in the Appendix.

5.2 Arduino NANO

The Arduino NANO board is situated on the hand strap, where there is an ADXL335 accelerometer, a MPL115A2 pressure sensor, and a RF transmitter. There are a total of 4 pins needed, where the accelerometer requires 2 input pins, the transmitter reqruires 1 onput pin, and the squeeze ball made with a MPL115A2 pressure sensor inside that requires 1 input pin. The main reason to have selected Arduino NANO as the slave microcontroller is due to its compact size. Since the microcontroller will be equipped on the wearable device, light-weight and compact-size are highly prioritized. A full diagram of the Arduino NANO pin assignment is also attached in the Appendix.

5.3 Encoder Interrupts

We need to properly configure the six digital pins on the Arduino MEGA to be hardware interrupt pins. The command `attachInterrupt(ChannelA,ISR,RISING)` is used to attach the interrupt activated on the rising edge of the pin. In addition, a critical section is set up to avoid unintentional errors in encoder readings. A sample ISR for one of the three encoders is provided as the following.

```
void ISR() {
    cli(); //stop interrupts
    // add 1 if ClockWise Rotation
    if (digitalRead(ChannelA) && !digitalRead(ChannelB)) {
        count++ ;
    }
    // subtract 1 if CounterClockWise Rotation
    if (digitalRead(ChannelA) && digitalRead(ChannelB)) {
        count-- ;
    }
    sei(); //restart interrupts
}
```

5.4 Sensor Signal Analog-to-Digital Conversion

The ADXL335 accelerometer used in the design has 3 analog outputs for each axis i.e. X, Y & Z; therefore, an ADC is required to convert these signals for further processing. Note that the ADXL335 uses a voltage standard of 3.3V; the manufactured PCB module has a 5V-to-3.3V converter which takes care of the input voltage conversion. However, we still need to keep in mind that the reference voltage is 3.3V while interpreting the 3 output signals from the module. As a result, the AREF pin on the Arduino is connected to 3.3v to set it as the reference.

From the embedded software level, we will use `analogReference(EXTERNAL);` to allow the hardware to realize the external reference voltage for reinterpreting incoming signals. Moreover, `analogRead()` function is called to ask the Arduino to read an analog signal and convert to a digital value ranged from 0 to 1023 (10-bit resolution).

6 Control Mechanism

The claw machine consists of three controllable stepper motors. All of them are placed together. Two of them are responsible for x-y plane motions, and the other one is responsible for releasing/retrieving (i.e vertical motion) the claw. Due to the fact that the user is only allowed to control the x-y plane position of the claw, in order to follow the user's hand motion in that plane, only two stepper motors are to be controlled by a closed-loop PID feedback mechanism. The third motor only needs to fully drop/withdraw the claw also preserving the mechanical components (i.e. prevent the stepper motor from turning over the limit).

6.1 Control Method Overview

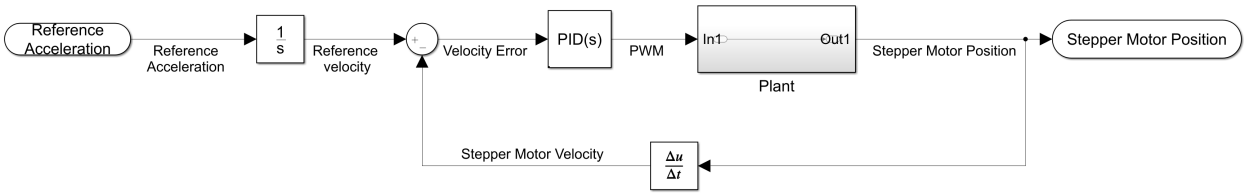

Figure 2: Closed Loop Control System for the Stepper Motors

A closed-loop feedback mechanism will be used to control 2 stepper motors in the X & Y direction separately. Figure 2 illustrates the closed-loop PID feedback control system for each stepper motor. The control methods will be the same for each of the motors but the input and output will be along X and Y axis. We will be replying on the 3 encoders equipped with the stepper motors as the output fed back which determines the error with the acceleration reference input provided by the accelerometer. We would like to introduce the PID controller, which takes account of the proportion, integral, and the derivative of the error between expected reference (interpreted from accelerometer on the user) and the actual output (from the encoder).

The main idea is that the X&Y stepper motors will work together to make the motion of the claw (i.e. velocity) follow the motion of the user's hand as much as possible. We are designing the system to be a Type 2 system (i.e. can follow a velocity input perfectly). As a result, the claw not only follows hand movements' directions, but the velocity as well, which makes the whole experience more immersive and entertaining.

6.2 Control Signal Analog-to-Digital Conversion

From Figure 2, there is one location where we need to convert the input signal type into a desired output signal type. The velocity error is fed into a PID controller and a PWM input to the plant is required. Thus, we need a ADC in between the PID controller and the plant. As earlier section has discussed, we will use `analogReference(EXTERNAL);` and `analogRead()` to handle this converting.

6.3 Control Gain Tuning

Note that the primary objective of the controller is to make the system a Type 2 system and stable with small overshoot and short rise time. Due to the nature of the design, we also would like the system to be sluggish and robust to external disturbances since rapid motion of motors with small movement of the sensor is not desired.

By using Matlab Simulink, we can simulate the system by giving a certain input and plotting the output using the scope block. It allows us to modify the PID gains and observe the changes in the output. Ultimately, we want the stepper motor velocity plot to be as close to the reference velocity plot as possible (given a certain overshoot, rise time, and zero steady state error). We then plot the Root-Locus Diagram to find the desired overall gain that satisfies the overshoot and rise time.

7 State Machine Diagram

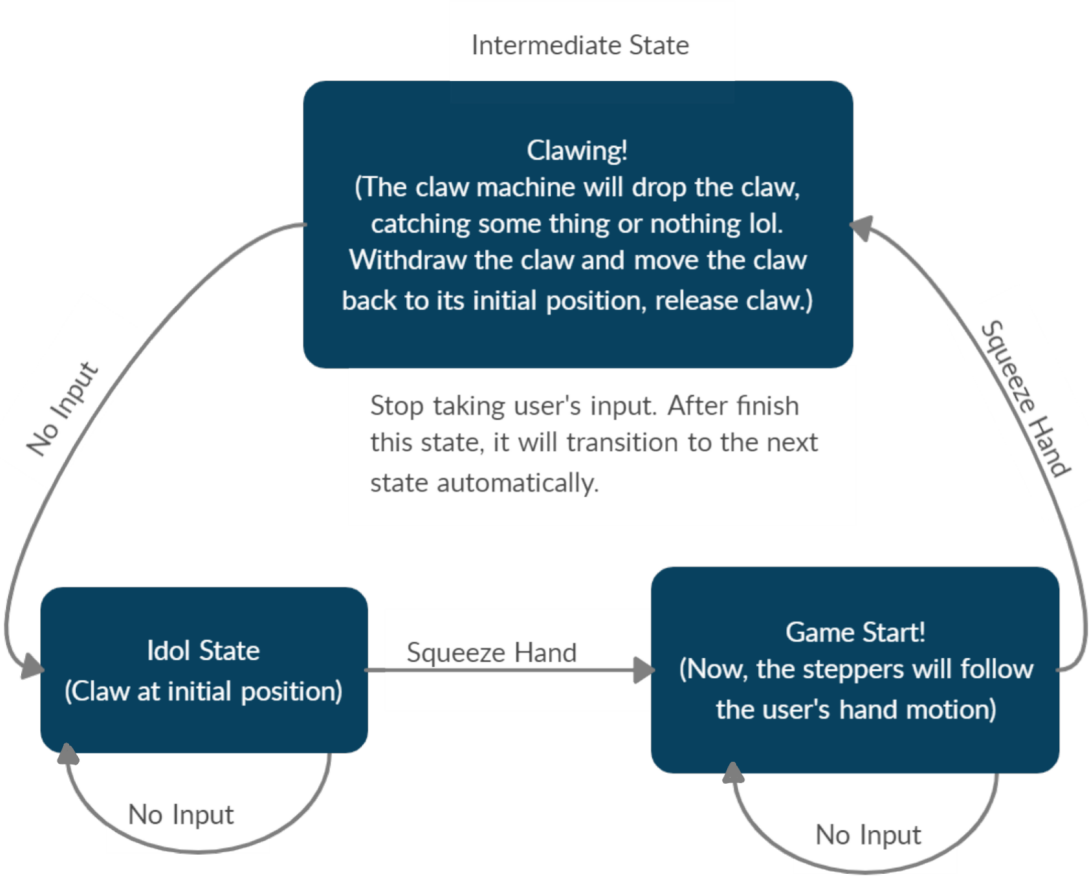

Figure 3: The State Machine Diagram for the Claw Machine

The flow diagram of the states is illustrated in figure 3. Three states are drawn and connected in a circle. Initially, the machine is at its idol state. The claw is at its initial position and the machine is waiting for user's input. Once the user squeeze the ball in their hand, the machine enters its next state. The game starts. Now, the user is able to move their hand and the claw will follow the motion. Once the user decides to drop the claw, they can simply squeeze the ball again to begin clawing. The machine state will enter the clawing state completing the task and going back to the idol state automatically.

8 Mechanical System

8.1 Gesture Controller

The originally proposed gesture controller is a "glove" device that users can wear on their hands to play. It includes an Arduino NANO microcontroller, an ADXL335 accelerometer, and a RF transmitter as described in section 4.2, as well as a battery pack for power. Figure 4 illustrates the design. The updated design, which is shown in figure 5, adds more interaction to the design by using a MPL115A2 pressure sensor inside a squeeze ball. The components are placed on a hand strap instead of a glove to account for handedness and improve usability overall.

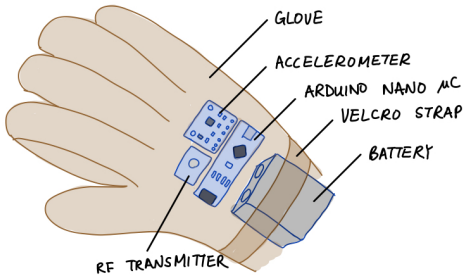

Figure 4: Proposed Gesture Controller

Figure 5: Updated Gesture Controller

8.2 Claw Crane Machine

Since the focus of this project is to achieve gesture control using microcontrollers, the mechanical aspect of the claw machine will not deviate much from traditional machines except having a smaller size. Figure 6 represents the high-level design of the claw machine with main components labelled. The machine would have a footprint estimated at 45 cm times 45 cm for the ease making parts using laser cutting. This is also a size feasible for an arcade machine that will be put on a table.

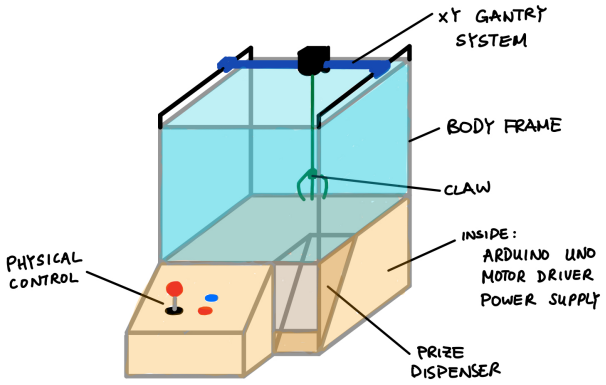

Figure 6: Claw Machine Design

9 Communication Protocol

We have considered using wired communication protocols like I2C and wireless approach, such as, IR and Bluetooth. With our primary objective to make the system more flexible and convenient for users, we have decided to continue using the wireless approach as mentioned in the proposal with minor changes. Details of the implementation will be discussed as the following.

9.1 Machine (Master Device)

In the main control loop, the device will constantly be checking if there is incoming IR signal. Then the signal will be decoded and used to calculate the stepper motor control input and adjust the motor position if needed.

9.2 Gesture Controller (Slave Device)

The slave device will retrieve the acceleration information every 100ms and send the acceleration in x and y direction through the IR transmitter.

9.3 Communication between Machine and Controller

We will be using the NEC protocol to send and receive IR signals between two devices. On the slave side, it will use `IRsend.sendNEC(data,32)` to send the information as a 32 bit number. Among this 32-bit number, the most significant 16-bit will be the acceleration detected in the x-direction, while the least 16-bit will be the acceleration in the y-direction.

10 Future Steps

Although we tried to be as extensive as possible in this conceptual design report, there are some steps that need to be done when the a physical system is built. The section outlines some of the critical steps in the process of building the physical system.

- Sensor Calibration: The sensors need to be tested for its intended operation. For example, a threshold value for the pressure sensor can be chosen for the squeeze ball to have 'ON' and 'OFF' states like a mechanical switch would. If the sensor has high amplitude noise causing 'switch bounce' issue during operation, a low-pass filter may need to be implemented to process the output.
- Tune PID Controller Further: Although the PID gains can be tuned using Matlab Simulink, as mentioned in section 5.3, the real-world components still differ from models. Depending on the situation, the gains need to be adjusted for optimal performance.
- Optimize for Control Delay: Compared to buttons, joysticks, or physical controllers in general, gesture control doesn't provide much feedback to users and is less intuitive to use. As a result, reducing delay between controller signal and target movement becomes critical for good user experience, and code optimization would be needed to reduce delay.

11 Appendix

Figure 7: A mini claw machine

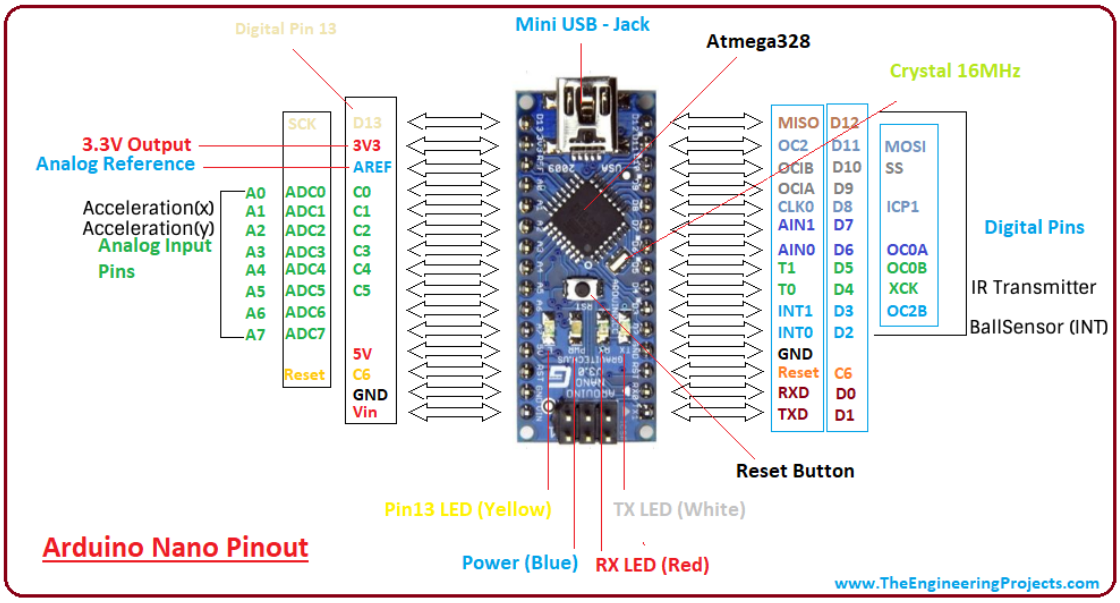

Figure 9: Arduino Nano Pinout[5]

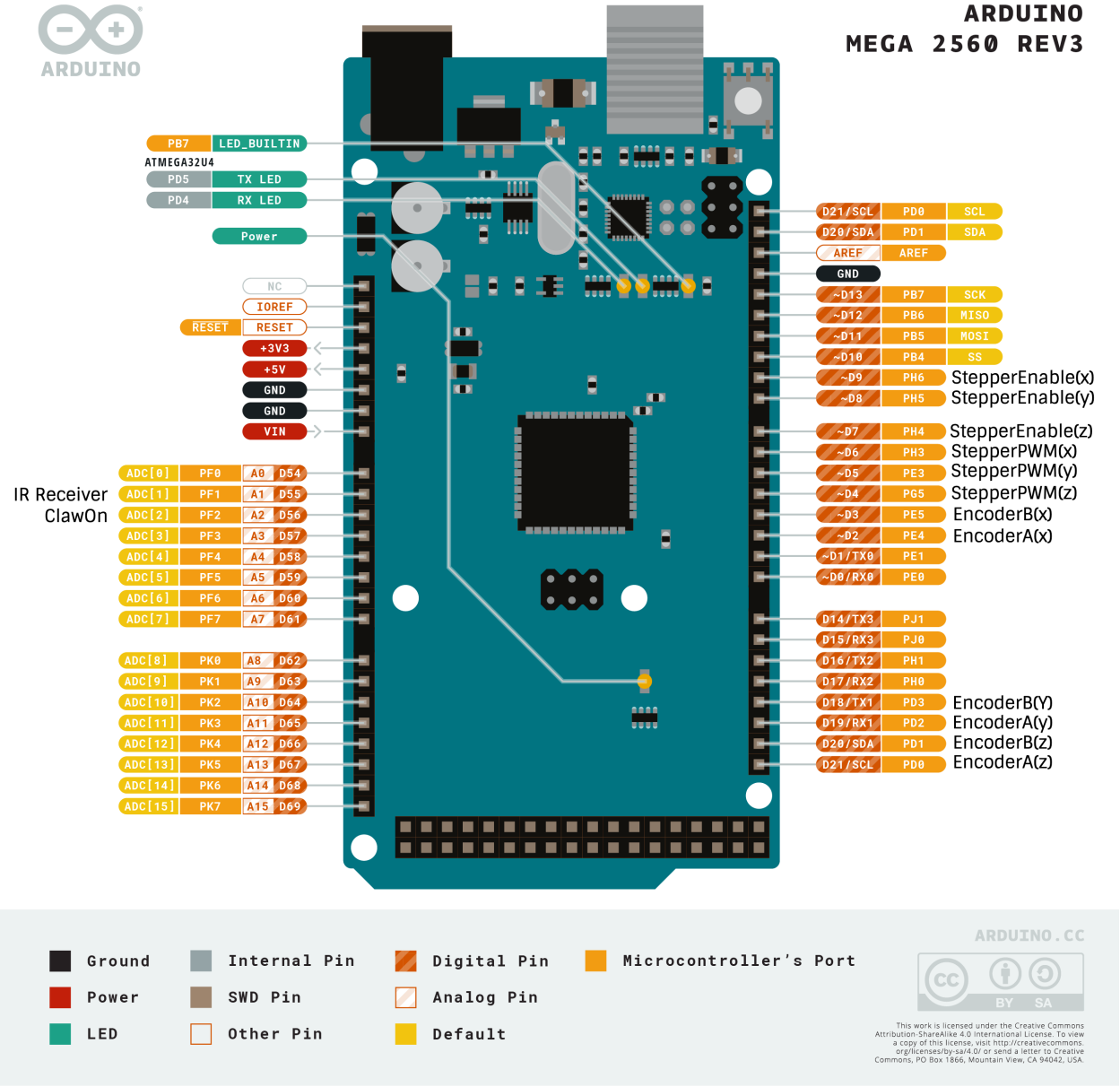

Figure 8: Mega Pin Assignment[4]

References

[1] "Claw crane," Jan 2020. [Online]. Available: https://en.wikipedia.org/wiki/Claw_crane

[2] C. P. Kumar and S. Madhu, "Gesture control robot with arduino," *IOP Conference Series: Materials Science and Engineering*, vol. 455, p. 012106, dec 2018. [Online]. Available: https://doi.org/10.1088%2F1757-899x%2F455%2F1%2F012106

[3] "Arduino digital pins with interrupts." [Online]. Available: https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/

[4] "Arduino mega 2560 rev3." [Online]. Available: https://store.arduino.cc/usa/mega-2560-r3

[5] A. Aqeel, "Introduction to arduino nano," Nov 2018. [Online]. Available: https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html