

스마트 분리수거

Auto Separate Collection Machine

안양대학교

전기전자공학과

김일하 유창호 우병영

스마트 분리수거

지도교수 김영렬

이 논문을 졸업논문으로 제출함

2022년 12월 7일

안양대학교

전기전자공학과

김일하 윤찬호 우병영

김일하, 윤찬호, 우병영의 졸업논문으로
인준함

심사위원 _____인

심사위원 _____인

심사위원 _____인

안양대학교

2022년 12월 7일

목 차

1. 서론

1.1 서론

2. 디바이스 구성

2.1 MCU

2.2 센서

2.3 모터

3. 시스템 설계

3.1 하드웨어

3.2 소프트웨어

4. 결론

4.1 시스템 제작

4.2 최종 동작

4.3 결론

4.4 부록

1. 서론

서론

캡스톤디자인 활동을 하면서 어떤 작품을 할지 회의를 하던 중 공원이나 길거리 등 공공장소에 있는 쓰레기통은 제대로 분리수거가 되지 않아 금전적인 측면과 환경적인 측면에서 피해를 많이 입고 있다는 기사를 접하게 되었다. 그래서 분리해서 넣을 필요 없이 알아서 분리수거를 해주는 쓰레기통이 있으면 피해를 줄일 수 있지 않을까? 라는 생각으로 고안하게 되었다.

스마트 분리수거기가 있으므로 기대되는 효과는 일반 쓰레기통과 다르게 버리는 즉시 분리수거를 도와주기 때문에 분리수거에 대한 부담이 줄어든다. 또한 분리수거가 활발히 이루어지게 된다면 재활용 용품들도 많이 나오게 되면서 경제적인 효과와 매립장 부지 절약 및 쓰레기를 다시 활용하기 때문에 환경보호 효과도 볼 수 있다.

우리가 목표하는 것은 쓰레기의 종류를 종이류, 캔류, 플라스틱류 이렇게 3가지로 구별하는 것을 목표로 하고 있다.

2. 디바이스 구성

2.1.1 AVR 마이크로컨트롤러

- 마이크로컨트롤러의 정의
- AVR 마이크로컨트롤러 개요
- ATmega128 마이크로컨트롤러

2.1.2 마이크로컨트롤러의 정의

마이크로 프로세서

- 프로세서를 하나의 칩 안에 집적하여 넣어 소형화한 형태
- 컴퓨터의 발전과 함께 고성능의 프로세서로 발전했으며, 최근에는 64비트의 고성능 프로세서들이 출시되고 있음.
- 마이크로 프로세서는 점점 고성능화하면서 범용 컴퓨터에 사용됨.

2.1.3 마이크로 프로세서의 발전

- 1971년 Intel사 4bit 마이크로프로세서 4004 개발
- 이후 여러 회사에서 8bit 마이크로프로세서 개발
Intel : 8008('72), 8080('74), 8085('76)
Motorola : MC6800('74), MC6805('76), MC6809('77)
Zilog : Z80('76)
- 이후 8, 16, 32, 64bit 마이크로프로세서 개발
Intel : 80186, 80286, 80386, 80486, Pentium...
Motorola : 68000, 68020, 68040, 68060...

2.1.4 마이크로컨트롤러(MCU : Micro Controller Unit)

- 지능화와 소영화를 위하여 마이크로프로세서에 메모리와 각종 주변장치들을 함께 집적하여 넣은 칩
- 마이크로프로세서 코어, 여러 가지 크기와 다양한 종류의 메모리, 여러 종류의 주변장치, 여러 종류의 입출력 보트를 하나의 칩에 집적
- 여러 응용 분야에 필요한 주변기기들을 한 칩 내에서 모두 제공

2.1.5 마이크로컨트롤러의 발전

- 1975년 : Texas Instrument TMS1000 개발(마이크로컨트롤러의 시초)
(1971년 Intel 4bit 마이크로프로세서 4004 개발)
- 1976년 : Intel 8bit M/C 8048(MCS-48) 개발
Motorola 8bit MC6801 개발
- 1980년 : Intel 8bit M/C 8051(MCS-51) 개발
- 1982년 : Intel 16bit MCS-96 개발
- 1988년 : Intel 32bit M/C 80960 개발

- etc.

2.1.6 마이크로컨트롤러(MCU)의 특징

- 주변장치들을 센싱 및 제어하기 위한 I/O 능력이 강화
- 타이머/카운터, 통신포트 내장 및 인터럽트 처리 능력 보유
- Bit 조작 능력이 강화
- 제품의 소형화 및 경량화
- 제품의 가격이 저렴(부품비, 제작비, 개발비 및 개발시간 절감)
- 융통성 및 확장성이 용이(프로그램만 변경)
- 신뢰성이 향상(부품 수 적어 시스템 단순, 고장율 적고, 보수편리)

2.1.7 마이크로컨트롤러(MCU)의 응용

- 산업: 모터 제어, 로봇 제어, 프로세스 제어, 수치 제어, 장난감 등
- 계측: 의료용 계측기, 오실로스코프 등
- 가전제품: 전자레인지, 가스오븐, 전자밥솥, 세탁기 등
- 군사: 미사일 제어, Torpedo 제어, 우주선 유도 제어 등
- 통신: 휴대폰, 모뎀, 유무선 전화기, 중계기 등
- 사무기기: 복사기, 프린터, plotter, 하드디스크 구동장치 등
- 자동차: 점화 타이밍 제어, 연료 분사 제어, 변속기 제어 등
- 생활: 전자시계, 계산기, 게임기, 금전등록기, 온도조절기 등

2.1.8 마이크로컨트롤러(MCU) 제조사

- Motorola(FreeScale) : MC6805, MC68HC16, MC68332, HCS12
- Samsung : KS51, KS88, KS16, KS32
- Microchip : PIC16/17
- Atmel : AVR시리즈, 8051시리즈
- Zilog : Super-8
- Stmicro : STM32 시리즈
- Texas Instrument : MSP시리즈
- etc.

2.1.9 마이크로컨트롤러(MCU)의 발전 방향

-고성능화

32비트 ARM 코어를 내장한 고성능 마이크로컨트롤러

-다기능화

다양한 특수 기능들을 내장한 다기능 마이크로컨트롤러

-소형화

초소형 임베디드 시스템 장착을 위한 초소형 마이크로컨트롤러

-저전력화

소형 배터리로 장시간 동작이 가능한 초저전력 마이크로컨트롤러

-저가격화

1\$ 이하의 가격을 가지는 저가격 마이크로컨트롤러

2.1.10 AVR 마이크로컨트롤러

-ATMEL사가 1997년에 처음 발표한8비트 제어용 마이크로프로세서

-Alf-Egil Bogen과Vegard Wollan의 진보된RISC 기술을 기반되었다고 하여 첫 글자를 따서AVR이라고 명명

-많은 장점 때문에 상대적으로 늦게 출시되었음에도 불구하고 8051이나PIC을 능가하는 인기를 단시간내에 얻게 되었음.

2.1.11 AVR 마이크로컨트롤러의 특징

- RISC (Reduced Instruction Set Code) 구조

- 하버드 아키텍처(Harvard Architecture)

- 32개의8비트 범용 레지스터를 가지는 레지스터 중심형 구조

- CMOS 기술 채택으로 소비전력이 매우 적고 동작전압이1.8-5.5V로 큼

- 다양한 동작 모드를 제공해 저전력 동작 지원

- 1K-256Kbyte 플래시롬, EEPROM 및SRAM이 작은 칩 하나에 내장.

- 유사RISC 구조와32개의 레지스터 사용과 고집적으로 1MHz당1MIPS의 처리속도.

- 8핀에서100핀의 외형과 이에 상응하는 메모리와 기능을 갖는 다양한 시리즈의 제품이 존재

- 모델에 따라 매우 다양한I/O 기능을 사용 가능

- 외부 시스템 버스를 이용하여 데이터 메모리 또는I/O 디바이스를 확장 가능.

- 다양한 인터럽트 소스와 처리 기능 보유.

- 무료 개발 도구인AVR Studio등, 다양하고 값싼 개발 도구 제공.

- ISP(In System Program) 기능 제공.

2.1.12 AVR 마이크로컨트롤러의 종류

- Tiny시리즈

1. 핀수가 8-24핀 정도의 작은 외형
2. 대부분 외부 시스템버스가 없음
3. 작은 메모리 용량(내부에 1K-2K byte 정도의 플래시 메모리를 보유)
4. 저속 저성능
5. 저가격(소형제어기에 적당).

- Mega 시리즈

1. 28-100핀 정도의 외형
2. 큰 메모리 용량(내부에 8K-256Kbyte 정도의 플래시 메모리와 256-4K byte 정도의 EEPROM과 512-4K byte의 SRAM을 내장)
3. 고속 고성능(20MHz의 클럭에서 20MIPS의 속도, 다양한 기능)
4. 고가격.

- AT90 시리즈

1. 중간 정도 사양을 가진 시리즈로 현재는 널리 사용되지 않음.

2.1.13 ATmega128 마이크로컨트롤러

- Atmel사의 8-bit 마이크로컨트롤러
- 고기능의 AVR Mega 시리즈 중에서도 가장 고성능 대용량의 마이크로컨트롤러임.



2.1.14 ATmega128 사양

- 고성능 저전력 AVR 8bit Microcontroller
- 향상된 RISC 아키텍처(16MIPS @ 16Mhz)

- 133종 명령세트(in 1 cycle)
- 32 x 8bit 범용 레지스터+ 주변 컨트롤 레지스터/Multiplier (in 2 cycle)
- 16MHz에서 16MIPS progress
- 비휘발성 프로그램과 데이터 메모리
- 128K Byte 내부 프로그램 가능한 ISP Flash memory
- 선택적인 Boot code section (used In-System Programming by On-chip Boot

Program)

- ISP (In System Programming)를 통해 어플리케이션 영역과 부트영역에 있어 F/W 다운로드 가능

- 4K Byte EEPROM
- 4K Byte 내부 SRAM
- 외부 최대 64K 메모리공간
- 프로그램 Lock-ISP을 위한 SPI 인터페이스
- 내장 메모리 Programming과 On-Chip Debug를 위한 JTAG 포트 지원

2.1.15 ATmega128 주변장치 특징

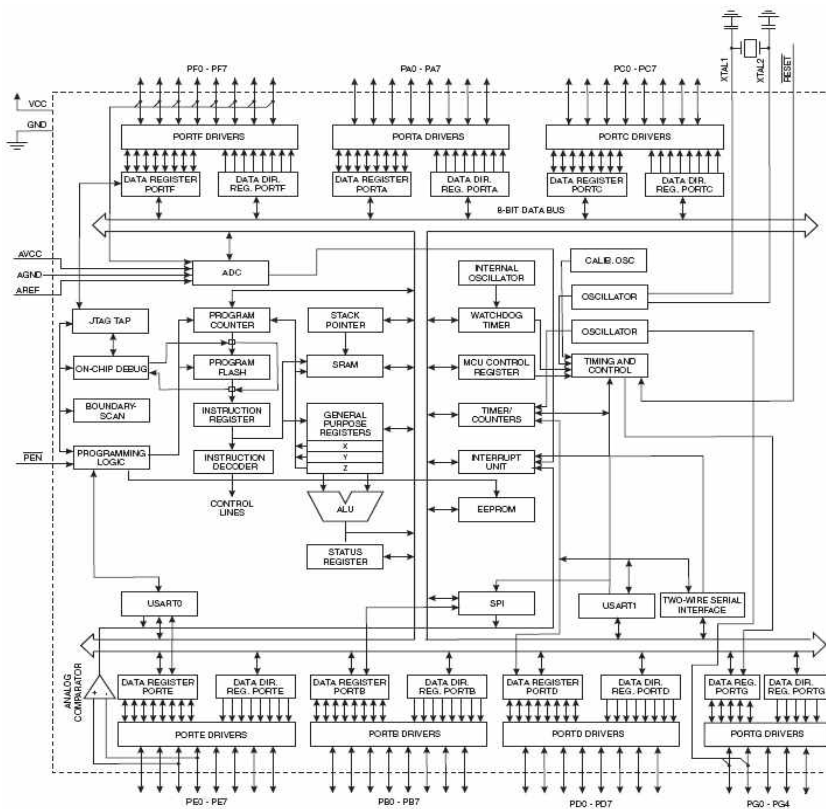
- 2개의 8비트 타이머/카운터(Timer/Counter)(0,2), 2개의 16비트 타이머/카운터(1,3)
- 2개의 8bit PWM 채널
- 6개의 프로그램 가능한 2~16bit PWM 채널
- 분리된 프리스케일러와 비교모드, 캡처모드를 가진 두개의 확장 16비트 타이머/카운터
- 분리된 오실레이터에 의한 Real Time Count
- Output Compare Modulator
- 8 채널, 10bit ADC
- Two-wire Serial 인터페이스
- 두개의 시리얼 UART
- Master/Slave SPI 시리얼 인터페이스
- 프로그램 가능한 워치독(Watchdog) 타이머
- 아날로그 비교기

2.1.16 ATmega128 마이크로컨트롤러 코어(Core) 특징

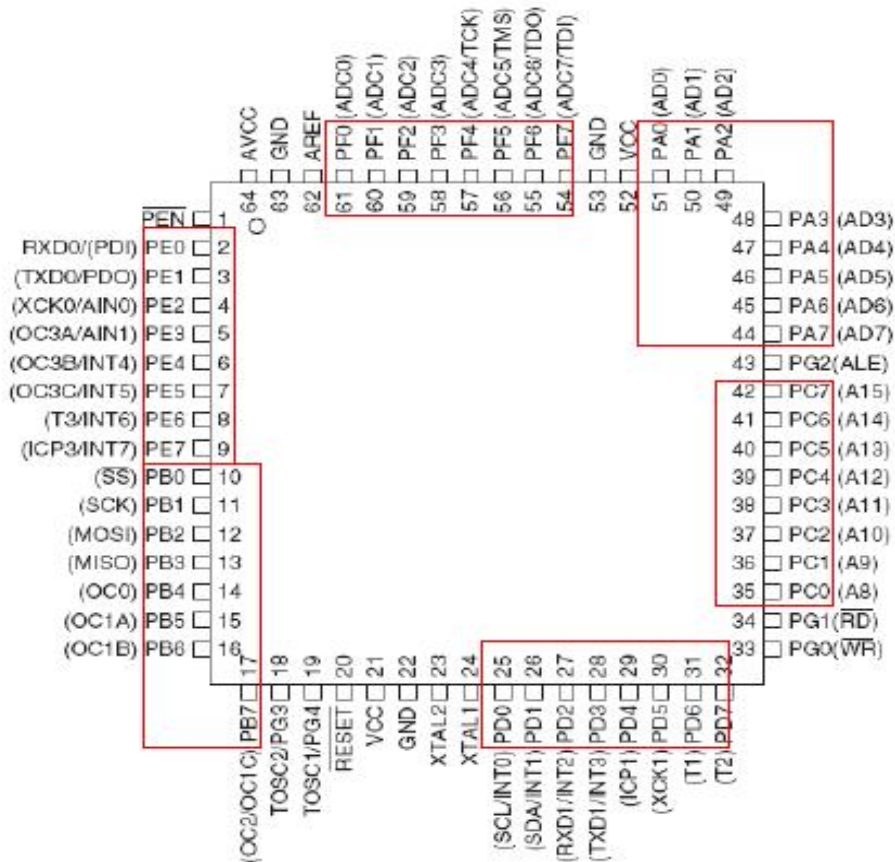
- Power-on Reset, 안정된 전원공급을 위한 Programmable Brown-out Detection
- 내부 RC 오실레이터

- 외부와 내부의 인터럽트 소스
- 6개의 Sleep 모드
- 소프트웨어적으로 선택 가능한 클럭 주파수
- 전체 플립Disable* I/O 와Package
- 프로그램 가능한53개의I/O
- 64-lead TQF / 64-pad MLF* Operation Voltage
- 2.7 ~ 5.5V : ATmega128L* 속도- 0 ~ 8 MHz : ATmega128L

2.1.17 ATmega128 내부 구조



2.1.18 ATmega128 외형과 핀기능



2.1.19 ATmega128 패키지

- 64pin
- TQFP 혹은 MLF
- 6개의 범용 입출력 포트 제공

2.1.20 제어신호

- RESET(핀20): 시스템 리셋신호.
- XTAL1, XTAL2(핀24,23) : 발진용 증폭기 입력 및 출력 단자.
- Vcc(핀21,51): 전원 입력 단자.
- GND (핀22,53,63) : 그라운드 입력 단자.

- AVCC(핀64): AD변환기 및 포트F에 대한 공급 전압
- AREF(핀62): ADC 참조 전압(Reference Voltage)
- PEN(핀1): ProgramingEnable 신호. 파워 온 리셋시 0상태로 유지해 SPI를 허용하게 한다.

2.1.21 AVR 마이크로컨트롤러 입출력 포트

- 6개의8비트I/O포트와1개의5비트I/O포트로 구성.
- 출력포트의 버퍼는 많은 유입전류와 유출전류를 사용(최대40mA)할 수 있음.
- 모든 포트 핀은 개별적으로 내부 풀업 저항을 사용할 수 있음.
- 모든I/O핀은VCC와GND사이에 다이오드를 접속하여 포트를 보호.
- Read-Modify-Write기능을 가지고 있어, 비트 단위의 포트 설정이 가능.
- 각 포트에 대한 데이터 출력용 레지스터(PORTx)와 데이터 입출력 방향 지정용 레지스터 (Data Direction Register: DDRx), 그리고 데이터 입력용 레지스터 (PINx)를 보유.

2.1.22 입출력 포트 제어용 레지스터

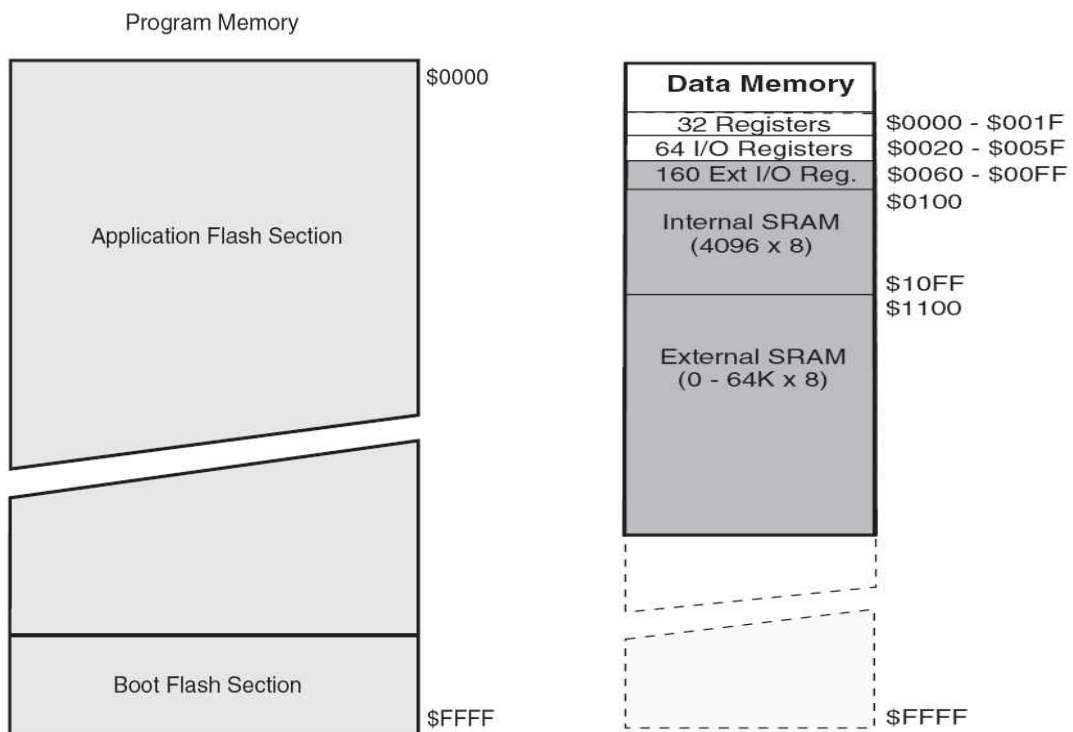
- DDRx 레지스터
입출력의 방향설정을 하기 위한 레지스터.
DDRA~DDRG레지스터의 해당 비트에 '1' 을 쓰면 출력, '0' 을 쓰면 입력으로 설정.
- PORTx 레지스터
데이터를 출력하기 위한 레지스터이다
출력을 원하는 데이터 값을 PORTx 레지스터에 넣어주면 된다
- PINx 레지스터
데이터 입력용 레지스터이다
PINx 레지스터에 해당하는 값을 읽으면 해당 핀의 값이 읽어진다.
- SFIOR 레지스터
Special Function IO Register.
AVR 입출력 포트의 특수 기능을 제어하기 위한 레지스터
SFIOR의 비트2(PUD: Pull-Up Disable)를 '1' 로 세트하면 풀업 저항을 비활성화 시킨다.

2.1.23 범용 입출력 신호

- 포트A (PA7~PA0:핀44~51) : 8비트 양방향 입출력 단자. 외부메모리를 둘 경우에는 주소버스(A7~A0)와 데이터버스(D7~D0)로 사용.

- 포트B (PB7~PB0:핀10-17) : 8비트 양방향 입출력 단자. SPI용 단자 혹은PWM 단자 로도 사용
- 포트C (PC7~PC0:핀35-42): 8비트 양방향 입출력 단자. 외부메모리를 둘 경우에는 주소버스(A15-A8)로 사용.
- 포트D (PD7~PD0:핀25-32): 8비트 양방향 입출력 단자. 타이머용 단자 혹은 외부 인터럽트용 단자로 사용.
- 포트E (PE7~PE0:핀2-9): 8비트 양방향 입출력 단자. 타이머용 단자, 외부인터럽트, 아날로그 비교기, USART용 단자로 사용.
- 포트F (PF7~PF0:핀54-61) : 8비트 양방향 입출력 단자. AD변환기 혹은JTAG 인터 페이스용 단자로도 사용.
- 포트G (PG4~PG0:핀19, 18, 43, 34, 33): 5비트 양방향 입출력 단자. 외부 메모리 스트로브 신호, RTC 타이머용 발진기 단자로 사용.

2.1.24 ATmega128 메모리 구조



- 하버드 구조를 가짐
- 프로그램 메모리: 프로그램 코드를 저장하고 실행시키기 위해 필요한 메모리
- 데이터 메모리 : 프로그램 실행에 필요한 데이터를 저장하는 메모리

RAM

- 레지스터
- SRAM

ROM

- EEPROM

외부 데이터 메모리

2.1.25 프로그램 메모리

- 프로그램 코드를 저장하고 실행하기 위해 필요한 메모리
- 내부128KB(byte)에 해당되는 플래시 메모리
- 각 번지가16bit씩 이뤄져 있음.
- Boot flash Section과 Application flash Section으로 나뉘어져 있음

(Self-Programming 지원)

1. Application flash Section : 프로그램 코드를 저장하는 공간
2. Boot Flash Section : Boot Loader HEX코드를 저장하는 공간
3. BootLoader를 이용하여 ISP 없이도 소프트웨어 업그레이드 가능

2.1.26 데이터 메모리(레지스터)

- 프로그램 이실행될 때임시로 데이터를 저장하는 고속 메모리
- 2가지 종류로 나뉘어 짐

범용 레지스터(General Purpose Register)

- ALU 연산에 필요한 데이터들을 임시로 저장하는 데 사용

특수 기능 레지스터(Special Function Register)

- 칩의I/O 제어나 상태 보고 등의 특별 기능들을 수행

2.1.27 범용 레지스터

- R0~R31까지 32개의 범용 레지스터를 보유

	7	0	Addr.	
General Purpose Working Registers	R0		\$00	
	R1		\$01	
	R2		\$02	
	...			
	R13		\$0D	
	R14		\$0E	
	R15		\$0F	
	R16		\$10	
	R17		\$11	
	...			
	R26		\$1A	X-register Low Byte
	R27		\$1B	X-register High Byte
	R28		\$1C	Y-register Low Byte
	R29		\$1D	Y-register High Byte
	R30		\$1E	Z-register Low Byte
	R31		\$1F	Z-register High Byte

2.1.28 특수 기능 레지스터

- 칩의 I/O 제어나 상태보고 등의 특별 기능들을 수행
- 2가지 종류로 나뉘어 짐

I/O 레지스터

- 64바이트(0x20에서0x5f번지) 로 구성
- 내장된 각종 I/O 장치를 제어
- 상태 레지스터(SREG) : ALU의 연산 후 상태와 결과를 표시
- 스택 포인터(SP) : 스택 위치 표시

- 확장 I/O 레지스터

160바이트(0x60에서0xff번지)로 구성

Atmega128에 추가된 각종 I/O를 제어

2.1.29 데이터 메모리(내부 SRAM)

- 4K의 SRAM이 내장(메모리 번지는\$0100~\$10FF까지 포함)
- 내부 프로그램 코드에 의해 발생하는 데이터를 일시적으로 저장

- 프로그램에서 사용자 변수의 저장영역이나 스택 영역으로 사용
- 일반 모드와 ATmega103호환 모드로 나뉘어 짐

2.1.30 데이터 메모리(EEPROM)

- 전원이 꺼지더라도 지속적으로 그 값을 유지해야 할 필요가 있는 별도의 데이터를 저장하기 위해 사용되는 메모리
- 총 용량4KB(byte)로 구성

2.1.31 외부 데이터 메모리

- 0x1100-0xffff번지에 외부 데이터 메모리를 연결하여 사용 가능
- 외부 램, 외부 플래쉬 롬 또는LCD나AD변환기와 같은 주변장치의 인터페이스용으로 사용 가능
- 외부메모리 인터페이스 기능
 1. 주변장치와 적절한 인터페이스를 위한0-3의 대기 사이클 지정 가능.
 2. 2개의 섹터로 외부 데이터 메모리를 분할하고 이들에 독립적인 대기 사이클을지정할 수 있음.
 3. 16비트 주소의 상위바이트 중에서 필요한 갯수의 비트만을 주소 버스로 동작하게 할 수 있음.
 4. 데이터 버스의 신호들이 동작할 때 전류 소비량이 감소되도록BUS-keeper 기능을 설정할 수 있음.

2.1.32 ATmega128 클록 종류

- 다양한 소스에 의해 클록을 발생시키고 분배가 가능.
- 소비 전력을 절약시키기 위해 개별적으로 공급 차단 가능
- 입력 클록의 종류
 1. CPU 클록 : 범용레지스터, 상태레지스터, 데이터 메모리와 같은AVR의 핵심적인 동작과 관련된 클록.
 2. I/O 클록 : 타이머, SPI, USART 등I/O모듈 대부분에서 사용되는 클록
 3. 플래쉬 클록 : 플래쉬롬과의 인터페이스를 제어하는클록.
 4. 비동기 타이머 클록 : 외부32kHz 수정 발진기를 소스로 하는 비동기 타이머용 클록.
 5. AD변환기 클록 : AD변환기용의 클록.

2.1.33 ATmega128 클럭 발생원

- 5가지의 클럭발생원이 존재
- CKSEL3~0와SUT1~0, XDIV 레지스터를 이용하여 클럭발생원과 주파수를 설정
- 다섯 가지 클럭 발생원
 - 내부RC(디폴트 클럭) 발진기 : 내장된RC 발진기를 사용하는 경우
 - 외부RC 발진기 : 정밀한 타이밍이 요구되지 않는 용도로 외부에RC 소자를 접속한 발진회로를 사용.
 - 외부 수정 발진기 : 외부에 크리스탈 또는 세라믹 레조네이터를 사용하는 경우
 - 저주파 수정 발진기 : 외부에32.768kHz의 낮은 주파수 크리스탈을 사용하는 경우
 - 외부 클럭 : 외부 다른 보드(8051보드, PIC보드)등의 클럭을 가져와 XTAL1단자에 연결.

2.1.34 ATmega128 Sleep 모드

- 전원을 절약할 수 있는6가지의 다양한 슬립모드가 제공
- MCUCR(MCU Control Register)레지스터를 설정하여 모드를 선택하고, SLEEP 명령을 실행하여 슬립모드에 돌입하도록 함.
- 6가지 SleepMode
 1. Idle모드
 2. ADC noise reduction 모드
 3. Power-down 모드
 4. Power-save 모드
 5. Standby 모드
 6. Extended Standby 모드

2.1.35 ATmega128 Reset 모드

- 정상적으로 동작하고 있는 마이크로컨트롤러가 리셋되면 모든I/O레지스터값이 디폴트값으로 초기화되고, 프로그램은 리셋 벡터에서 시작된다.
- 리셋의5가지 발생원
 1. Power-on Reset : 전원전압VCC의 전압 레벨에 따른 리셋
 2. External Reset : /RESET 핀에 의한 외부 리셋
 3. Watchdog Reset : 워치독 타이머에 의한 리셋

4. Brown-out Reset : Brown-out Detector에 의한 리셋.
5. JTAG AVR Reset : JTAG 시스템에 의한 리셋.

2.2.1 적외선 센서

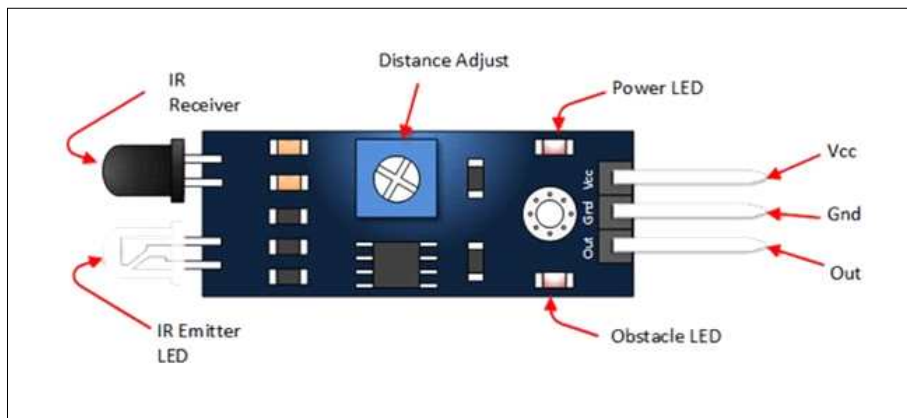
적외선 센서는 장애물 회피 센서 모듈 반사 광전 센서인 FC-51을 사용했다. FC-51은 적외선 발광부와 적외선 수신부가 존재한다. 적외선을 발광부에서 송신하고 물체에 반사되어 수광부에 수신되는 것을 감지하여 장애물 유무를 판단하는 센서 모듈이다. 사용전압은 DC 3 ~ 5V를 사용하며 감지 거리는 사용자의 설정에 따라서 2 ~ 30cm이하로 물체를 검출할 수 있다. 전위차계를 통한 감지 거리를 조절하는데 Distance Adjust를 시계방향으로 돌리면 감지 거리가 증가하며, 반 시계방향으로 돌리면 감지 거리가 감소 된다. 이 모듈이 갖는 특징으로는 광 센서로서 광 선제가 갖는 특징들을 가지고 있다. 특징들로는 빛의 양, 물체의 모양이나 상태 . 움직임등을 감지하는데 광을 발산하여, 광이 물체에 부딪혀 반사되어 오는 것을 받아들이고 그 물체의 움직임이나 빠르기를 알아내는 구도이다. 적외선 광에서 적외선 광까지의 광 파장 영역의 광을 검출하여 이것을 사용 가능한 양으로 변환하는 소자이고 수광한 광 에너지의 변화를 전기 신호(전기 에너지로)로 변환시키는 것이 광센서의 특징이다. 광센서의 특징이 아닌 적외선 센서의 특징으로는 적외선의 파장은 가시광보다 길고 전파보다 짧은 전자파의 일종으로 광기전력 효과를 이용한 포토 LDE와 포토 트랜지스터를 통칭하며 대략 0.78um에서 1.000um까지의 빛을 감지하는 센서를 말한다. 일정한 주파수의 빛을 발산만 하는 발광부 센서와 발광부에서 발산된 빛을 받아들이기만 하는 수광부 센서로 이루어져있으며 이들 센서는 사용에 있어서 특히 주의할 점이 극성이 존재한다는 것이다. 그러므로 사용 시 이들 극성에 따라 설계해야 한다는 특징을 가진다. 광전 효과를 이용하는 센서이기에 무접점으로 물체의 유/무를 판별이 가능하며, 최대 30cm까지 물체를 감지할 수 있다. 적외선 센서가 감지하여 출력하는 신호는 디지털 펄스 신호 (“H” , “L”)이다. 주변에서 적외선 센서가 사용한 곳을 찾아보면 디지털카메라, 자동문, 터치스크린, 침입 경보 시스템, 자동 조명 장치 등 물체의 감지가 필요한 곳에 사용되는 것을 볼 수 있다. 센서의 보드 크기는 3.2 x 1.4cm로 소형의 센서이다.

구 분	사 양
검출 거리	2 ~ 30cm
검출각	35°
작동 전압	3 - 5V DC
출력 유형	디지털 스위칭 출력 (0 및 1)

보드 크기	3.2 x 1.4cm
-------	-------------

[표 2-1] FC-51 사양

FC-51 적외선 센서의 구조는 [그림 2-1]와 같이 물체를 식별하기 위한 송/수신부와 거리 감지를 조절할 수 있는 Distance Adjust가 있다. 전원이 들어오면 표시를 해주는 LED와 물체가 감지되거나 감지가 되지 않았을 때 점소등이 되는 LED가 있다. 센서에 연결할 수 있는 단자는 3개로 되어있으며, VCC와 GND, 디지털 출력을 하는 OUT단자로 이루어져 있다. OUT단자에서 물체 감지가 되었을 때는 LOW(0) 신호를 주며, 물체 감지가 되지 않았을 때는 HIGH(1) 신호를 디지털 출력을 발생시켜 송신한다.

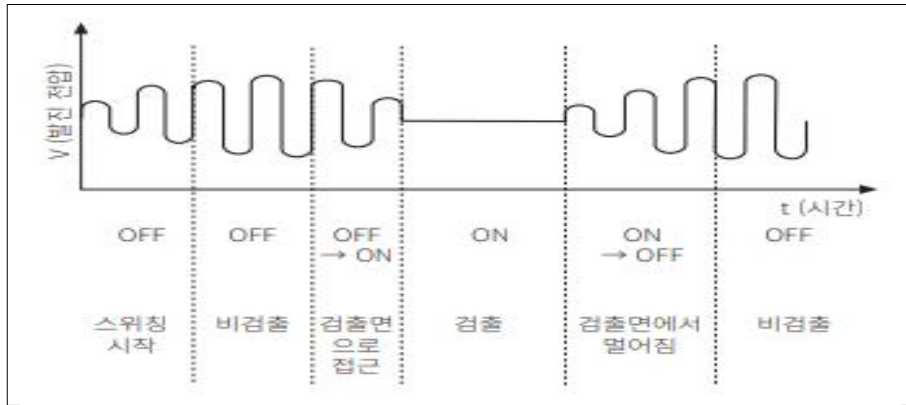


[그림 2-1] FC-51 구조

2.2.2 근접 센서

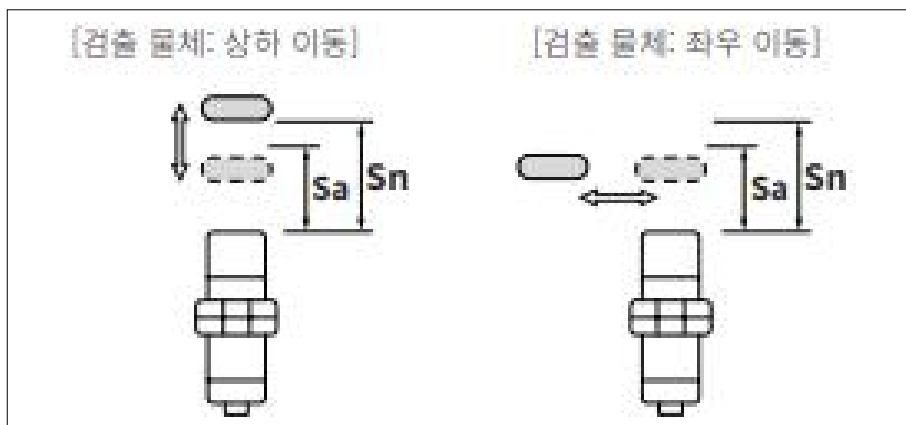
적외선과 같이 1차 분류에 쓰이는 센서로 PR18-8DP를 사용하였다. 근접 센서는 적외선으로 물체를 감지하고 금속/비금속을 탐지하기 위하여 쓰이는 센서로써 자성을 띄는 물체가 접근하면 HIGH(1) 신호를 출력하고, 자성이 없는 물체가 접근하면 LOW(0) 신호를 출력한다. 이 근접 센서는 2가지 사용 방법이 있는데, 고주파 발진형과 정전 용량형이 존재한다. 작품을 만들 때 있어서 적합한 것은 고주파 발진형이기에 고주파 발진형을 채택하여 사용하였다. 고주파 발진형의 원리는 검출 코일에서 발생하는 고주파 자계 내에 검출 물체 (금속 자성) 가 접근하면 전자 유도 현상에 의해 물체 표면에 유도 전류 (와전류) 가 흐르게 되어 검출 물체 내에 에너지 손실이 발생한다. 에너지 손실이 발생하게 되면 검출 코일에서 발생하는 발진 진폭이 감소 또는 정지하는데 발진 진폭의 변화량을 사용하여 검출 물체의 유/무를 판별하는 원리이다. 근접 센서의 고주파 발진형의 동작으로는 [그림 2-2]와 같이 전원을 인가하게 되면 약 60ms 이내에 전압의 진동

폭이 일정한 주파수대로 올라가게 되어 전기적인 자장이 형성된다. 이후 검출 물체가 접근하면 검출 물체 표면에 유도 전류가 증가하여 전압의 진동 폭이 작아지고 완전히 검출된 상태가 되면 0V에 가까워진다. 이런 미소한 전압의 진동 폭을 증폭시켜 출력부를 동작시킨다.



[그림 2-2] 근접 센서 고주파 발진형 동작 원리

근접 센서는 DC 3선식이고 검출 거리거리가 8mm로 자성 물체를 검출할 때 오차가 적게 나오게 위하여 검출 거리가 긴 센서를 채택하였다. 응차 거리는 검출 물체가 검출면으로 접근하여 출력이 동작한 지점에서 검출 물체가 검출면에서 멀어져 출력이 복귀한 지점 사이의 거리를 말하는데, 이 센서의 응차 거리는 검출 거리의 10% 이하이다. 센서에서 지정한 표준검출체는 25x25x1mm(철)로 지정했는데 철이 아니어도 자성만 띠고 있는 금속 물체면 검출이 된다. 설정 거리는 [그림 2-3]과 같이 외부의 영향 (온도, 전압, 기타 환경 등)에 의한 검출 거리 변동 요인을 포함하여 안정하게 사용할 수 있는 검출면과 표준 검출 물체까지의 간격을 말하며, 통상 정격 검출 거리의 70%가 되는데 센서의 설정 거리는 0 ~ 5.6mm이다.



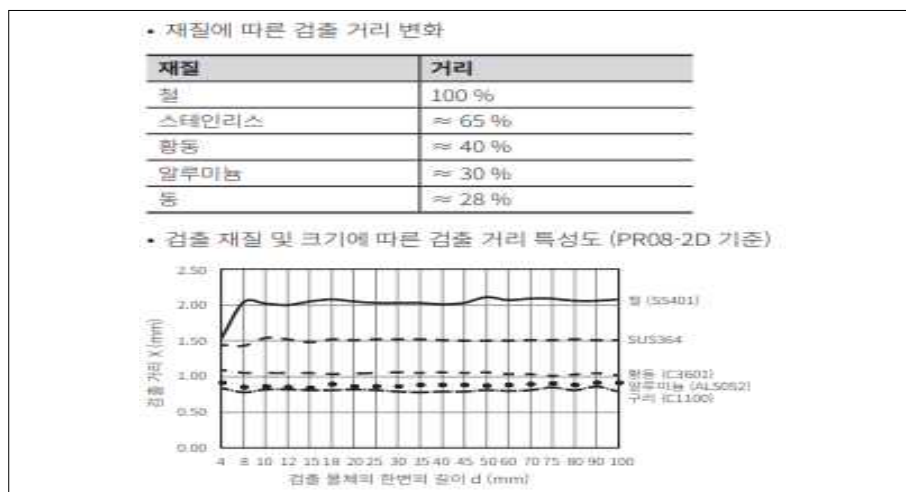
[그림 2-3] 근접 센서의 설정 거리요인

근접 센서의 전원 전압은 12 - 24V DC이며 소비전류는 10mA 이하이다. 자성 물체가 근접 센서의 검출 거리 내에서 검출이 되면 근접 센서 하단부의 LED에 불빛이 들어오게 된다. 근접 센서는 3개의 단자로 이루어져 있으며 [그림 2-4]와 같이 갈색 선은 VCC를 연결하는 선이며, 청색 선은 GND를 연결하는 선, 흑색 선은 출력을 연결하는 선이다. 근접 센서의 LED에 불빛이 들어오게 된다면 자성 물체 (금속)을 감지했기에 HIGH(1) 신호를 출력하게 되고, LED에 불빛이 들어오지 않는다면 자성 물체 (금속)을 감지하지 못하였기에 LOW(0) 신호를 출력하게 된다.



[그림 2-4] 근접 센서 단자

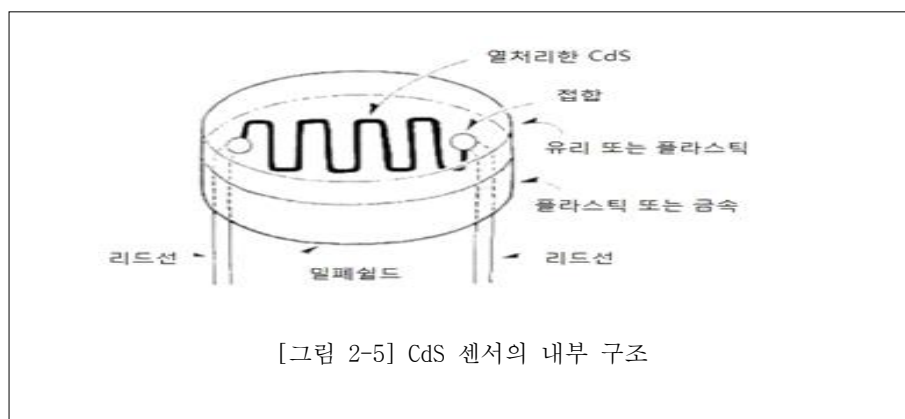
근접 센서는 표준검출체가 아니어도 물체에 자성만 띠면 센서가 감지한다. 하지만 검출 거리가 달라져서 감지하는데 오차 범위가 생길 수 있다. 금속 중에 비자성 금속인 알루미늄은 검출할 때 검출 거리가 많이 짧아져서 신경 써야 한다. 재질마다 검출 거리가 다른데 [그림 2-5]를 보면 철은 검출 거리가 100% 다 해당이 되며, 나머지 재질들은 각각 검출 거리가 짧아진다는 것을 볼 수 있다.



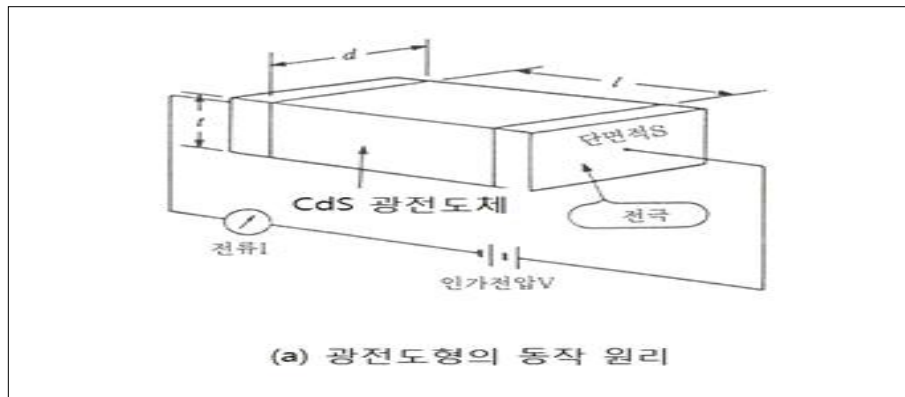
[그림 2-5] 재질에 따른 검출 거리 변화

2.2.3 조도 센서

2차 분류에 있어서 비금속인 플라스틱과 종이를 분류하기 위해 쓰이는 센서로는 GL5549 CdS Cell을 사용했다. 조도 센서가 갖는 특징을 2가지 이용하여 작품을 만들었다. 첫째로 갖는 특징으로는 조도 센서는 빛의 양에 따라 저항값이 변하는데 [그림 2-5]에 있는 열처리한 CdS이 광자의 양을 측정한다. 이때, 조도 센서의 크기, 지름이 클수록 빛을 측정하는 양이 더 많아진다. CdS란 황화카드뮴으로 가시광의 조도를 측정하는 센서로 가장 많이 사용되는 재료로 이 재료의 에너지 갭은 2.41 eV, 이에 해당하는 차단 파장은 0.52 μm 이다. CdS 조도 센서의 동작 원리는 광전도 현상을 이용한 센서이며 [그림 2-6]은 개념을 그림으로 표현한 것이다. 빛이 재료에 입사하게 되면 전자 - 정공 쌍이 생성되고, 생성된 전자는 운반자 농도의 증가를 가져와서 내부의 저항이 줄어들게 된다. 따라서 이러한 원리로 인해 빛이 많이 들어오면 저항이 점점 작아지게 된다. 또한 조도에 따른 저항 변화를 극대화하기 위해 [그림 2-5]와 같이 CdS 재료의 선로 저항을 올리기 위해 선폭은 좁게 만들고, 전체 길이는 지그재그 형태로 만들어서 저항을 크게 만든다. 빛이 없는 상태에서 저항이 커야만 측정할 수 있는 조도 범위가 넓어지게 된다. 즉, 주의가 밝을수록 광자 양은 많아져 저항값이 작아지고, 저항값이 감소되어 전류가 많이 흐르게 된다. 반대로 주위가 어두울수록 광자 양이 작아 저 저항값이 커지게 되고 전류가 적게 흐르게 된다. 주위 밝기와 저항값은 반비례한다는 것을 기억해야 한다.



[그림 2-5] CdS 센서의 내부 구조

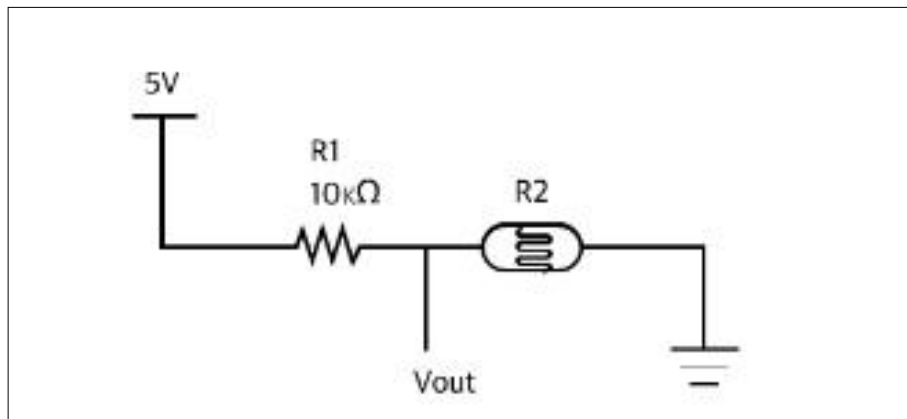


[그림 2-6] 광전도형의 동작 원리

조도 센서의 두 번째 특징으로는 극성이 없다는 특징이다. +극과 -극 부분이 정해져 있지 않다. 빛의 양에 따라 저항이 변하기에 가변저항 취급하기 때문에 저항 취급하면서 설계했다. 조도 센서는 최대 전압 150V DC를 가지며 최대 전력 90mW를 가진다. 피크 파장은 540nm을 가지며 명 저항은 50 ~ 150KΩ을 가지고 암 저항은 20MΩ을 가진다. [표 2-2]는 조도 센서의 규격이며 조도 센서를 이용할 때 [그림 2-7]과 같이 저항을 추가해줘서 전압분배법칙을 사용해줬다. [그림 2-7]과 같이 10KΩ을 추가해주는 이유는 조도 센서도 일종의 저항이고, 빛의 양이 매우 많은 경우 저항이 매우 작아져 과전류가 흐른다. 또한 조도 센서의 저항값에 따른 전압을 신호로 입력받기 위해 10KΩ 저항을 달아주는 것이다. 조도 센서에서 플라스틱을 투과한 빛을 받으면 저항값이 작아져 전압분배법칙을 통해 5V에 가깝게 나오기에 HIGH(1) 신호가 출력되고, 종이로 가려져 조도 센서가 빛을 받지 못할 때는 저항값이 높아져 0V에 가깝게 나와 LOW(0) 신호가 출력된다.

구분	사양
Model	GL5549
Vmax(VDC)	150
Pmax(mW)	90
Ambient temp(°C)	-30 ~ +70
Spectral peak(nm)	540
Light Resistance at 10Lux(KΩ)	50-150
Dark Resistance(MΩ)	20
Gamm a value at 100-10Lux	0.8

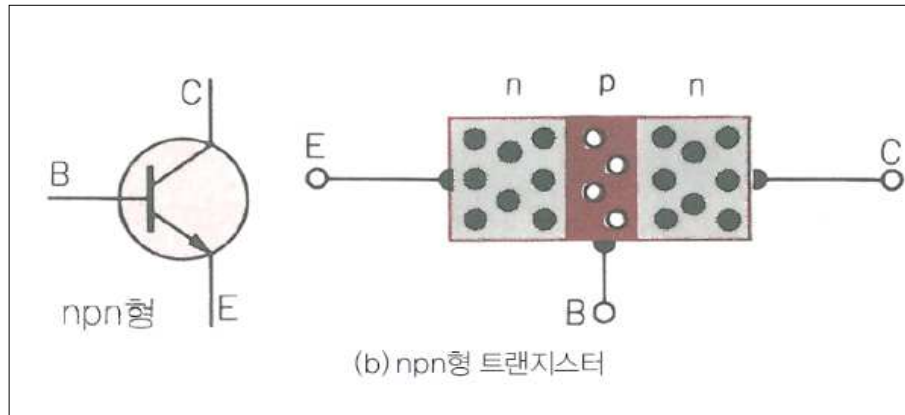
[표 2-2] GL5549 사양



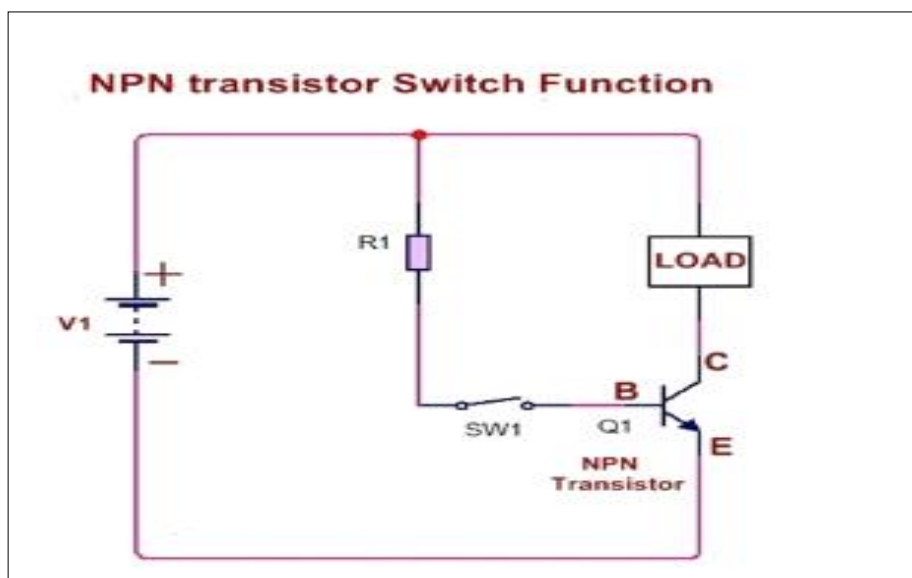
[그림 2-7] 조도 센서 전압분배

2.2.4 기타(트랜지스터, 커패시터, 저항, 발광 다이오드)

트랜지스터는 NPN형 2N2222A를 사용하여 스텝모터 작동을 위한 스위칭 기능으로 사용했다. 트랜지스터는 세 가지 기능으로 사용할 수 있는데 즉, 스위칭, 검파, 증폭용으로 모든 전자 시스템에 한가지 또는 여러 가지 형태로 사용된다. 작품에서 사용되는 NPN형 트랜지스터는 N 형, P 형, N 형의 반도체를 [그림 2-8]과 같이 접합하고 각 반도체로부터 도선을 내놓으면 NPN형 트랜지스터가 된다. [그림 2-8]을 보면 트랜지스터 내부에 화살표 표시가 되어있는 선이 있는 것이 이미터(E : Emitter)라고 불리는 단자이고 B라고 써있는 선이 베이스(B : Base), 나머지는 컬렉터(C : Collector)라고 불린다. 작품에서 쓰이는 기능으로는 스위칭 기능을 사용하는데 [그림 2-9]와 같이 NPN 회로에서는 베이스 전류 IB가 흐르지 않으면 컬렉터 전류 IC도 흐르지 않는 성질을 가지고 있다. 따라서 베이스 전류 IB를 ON, OFF 함으로써 컬렉터 전류 IC, 이미터 전류 IE를 ON, OFF 하는 방식이다. NPN형 트랜지스터를 채택하게 된 이유로는 스텝모터를 작동시킬 때 있어서 PNP형 트랜지스터의 캐리어는 정공이며 NPN형 트랜지스터의 캐리어는 전자이다. 전자의 이동성은 정공보다 빠르기 때문에 스텝모터의 코일을 활성화시켜주기 위한 고속 스위칭 작업에 적합하여 채택하게 되었다.



[그림 2-8] NPN형 트랜지스터 구조



[그림 2-9] NPN형 트랜지스터 스위칭 작용

커패시터는 설계된 회로의 전반적 잡음 제거 역할을 위해 용량 값 10uF과 22uF을 사용하였다. 커패시터는 보통 축전기라고 불리는데 축전기의 기능이 아닌 교류에서 직류를 차단하고 교류성분을 통과시키는 성질을 이용했다. 이 원리를 사용하면 불안정한 전원을 잡아주고 직류를 차단하여 교류를 통과시키게 되면 회로의 노이즈가 제거되며, 회로의 안정된 작동을 기대할 수 있다. 커패시터의 용량이 크거나 주파수가 높을수록 교류는 더욱 잘 통과하게 된다. 커패시터의 용량은 F(패럿)이고 하나의 극으로 이루어진 커패시터가 존재하며 극성이 있는 커패시터가 존재한다. 작품에서 사용한 커패시터는 전해 커패시터이기에 다리가 긴 쪽이 (+) 극, 다리가 짧은 쪽이 (-) 극이다.

저항은 물체에 전류가 흐를 때 이 전류의 흐름을 방해하는 성질을 가지고 있는 것이 저항이라고 한다. 저항의 단위는 옴(Ω)이라고 하며 옴의 법칙을 따랐을 때 전류와 반비례하고 전압과는 비례한 성질을 가지고 있다. 작품에서 저항 역할은

풀업 저항과 트랜지스터의 베이스 보호를 위해 사용되었다. 풀업 저항을 사용하는 이유는 플로팅 현상을 없애기 위하여 쓰는 것이다. 플로팅 현상이란 LOW(0)과 HIGH(1) 둘 다 아닌 상태를 말하며 아무것도 아닌 상태가 된다. 일반적으로 디지털 회로는 LOW(0)과 HIGH(1)의 전압이 인가되어야 하는데 플로팅 현상은 LOW(0)과 HIGH(1)의 중간전압을 갖기에 내부상태가 불안정하게 되거나 전력 소비가 증가하게 된다. 중간전압의 특징을 가지고 있기에 방해 전류 흐름이 생겨서 회로를 파괴할 수 있기에 안정적인 전압 인가를 위하여 풀업 저항을 사용하였다.

트랜지스터	커패시터	저항
		

[그림 2-10] 트랜지스터, 커패시터, 저항 사용 부품

발광 다이오드는 2차 분류에서 조도 센서로 빛을 비춰주기 위하여 쓰이는 센서로써 5파이 발광 다이오드 고휘도 LED를 사용하였다. 빛을 조도 센서로 전달하기 위한 장치로써 세기가 강한 5파이 LED를 채택하게 되었고 긴 다리가 (+) 극이며 짧은 다리는 (-) 극이다. 전압은 3 ~ 3.2V로 작품에 사용할 때는 저항을 추가해주어 발광 다이오드의 회로가 타는 것을 방지해주었다. 발광 다이오드는 플라스틱과 종이에서의 빛 투과량을 조도 센서로 전달하는 역할을 하게 된다.



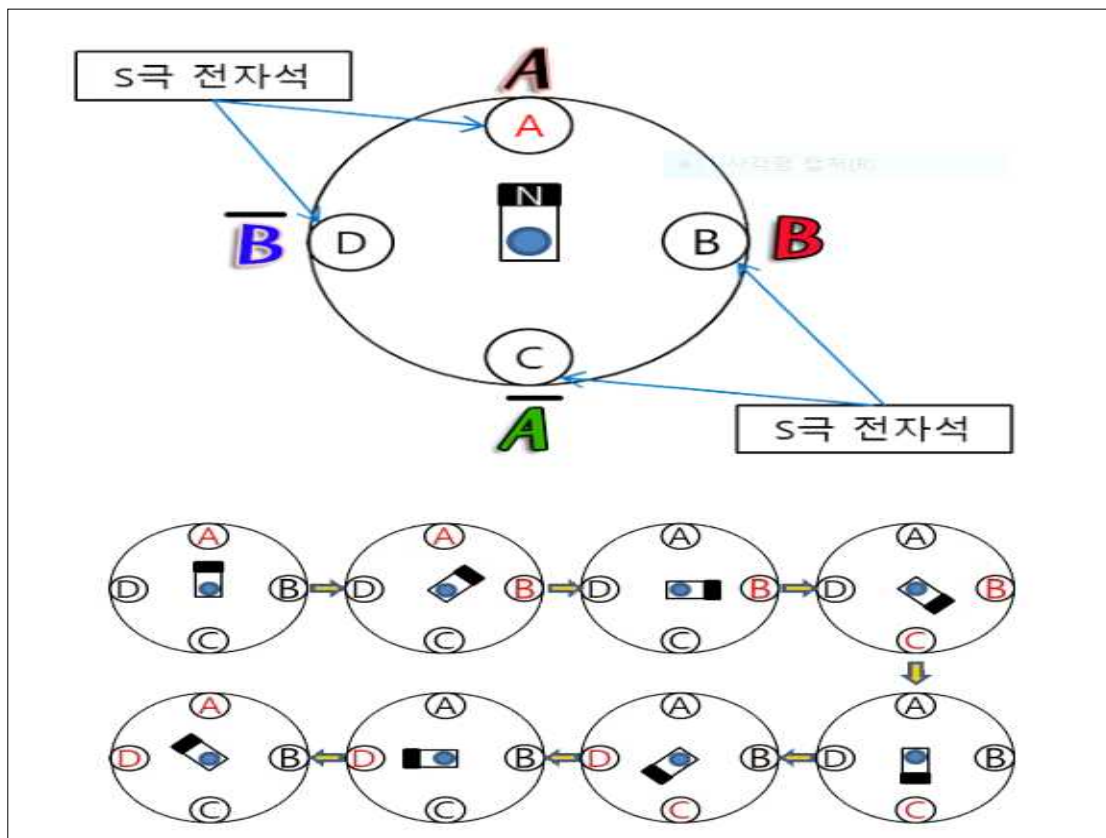
[그림 2-11] 발광 다이오드 구조

2.3.1 스텝모터

작품에서 움직여줄 모터를 선정함에 있어서 정확하고 힘이 강한 모터를 필요로 해서 스텝모터를 선정하게 되었다.

스텝모터는 기동, 정지, 정역회전의 응답성이 좋고 스텝당 각도오차가 적으며 오차가 누적되지 않는다. 그리고 디지털신호등의 펄스 입력에 오픈루프 제어가 가능해 시스템 구성 회로를 간단하게 할 수 있다는 장점이 있다.

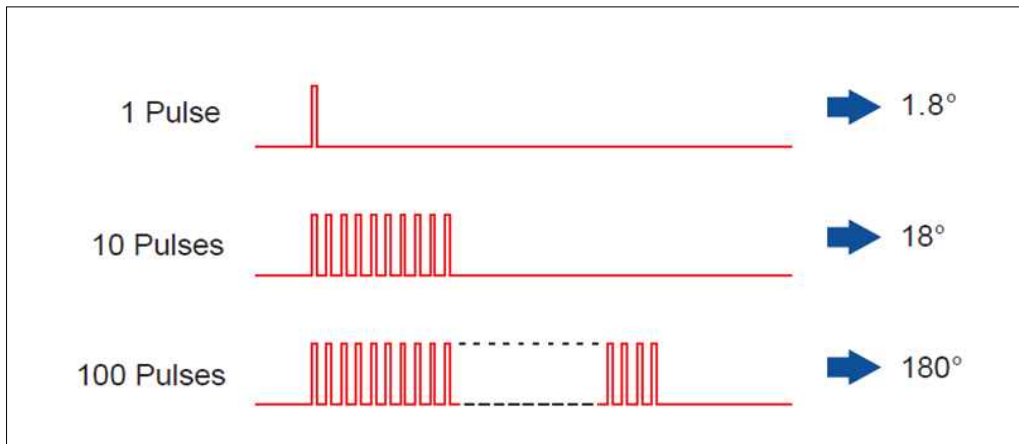
스텝모터는 내부의 축을 중심으로 여러 개의 코일들이 전자석들의 조각들을 형성하고 있다. 내부의 축을 중심으로 전자석들이 시계방향, 혹은 반시계방향으로 자석이 될 때 모터의 축이 자석이 된 전자석 방향으로 다가가기 위해 움직이면서 모터가 회전하게 된다.



[그림 2-12] 스텝모터의 회전

그림으로 설명하자면 스텝모터의 축에는 N극의 전자석이 있고, 주위에 전류를 흘려주면 S극이 되는 "A", "B", "C", "D" 의 전자석이 있을 때, 전류를 A - B - C - D 순으로 흘려주면 모터의 축이 움직이면서 시계방향으로 움직이게 되는 것이다.

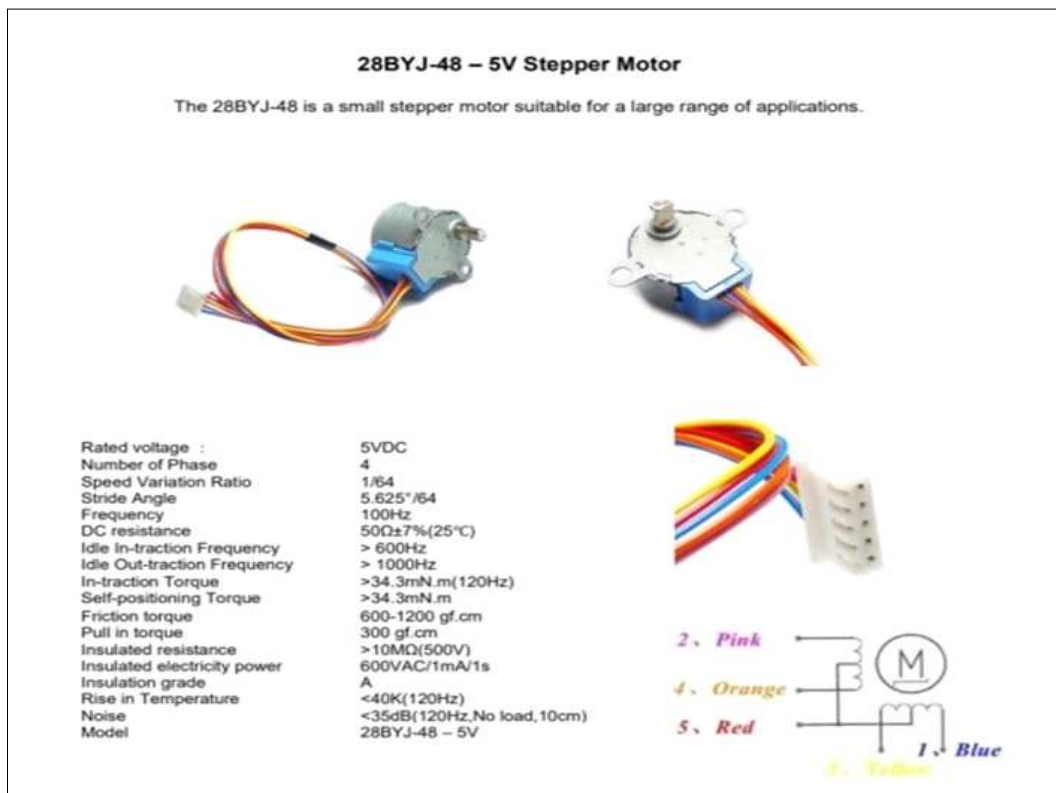
또한 스텝모터는 전류를 흘려주는 것으로 움직이는 것이 아니라 주어진 펄스에 따라 지정된 각도 따라 단계별로 움직이는 모터이다. 스텝모터마다 정해진 스텝각은 서로 다르고, 펄스 1개당 정해진 스텝각 1번 회전한다.



[그림2-13] 스텝모터의 스텝각

그림에서와 같이 스텝각이 1.8도인 스텝모터는 1펄스 당 1.8도를 회전한다면, 10펄스를 받았을 때는 18도를 회전하고, 100펄스를 받으면 180도를 회전하게 된다. 따라서 스텝모터의 스텝각을 알면 원하는 각도만큼 회전시킬 때 어느 정도의 펄스를 인가해주어야 하는지 알 수 있다.

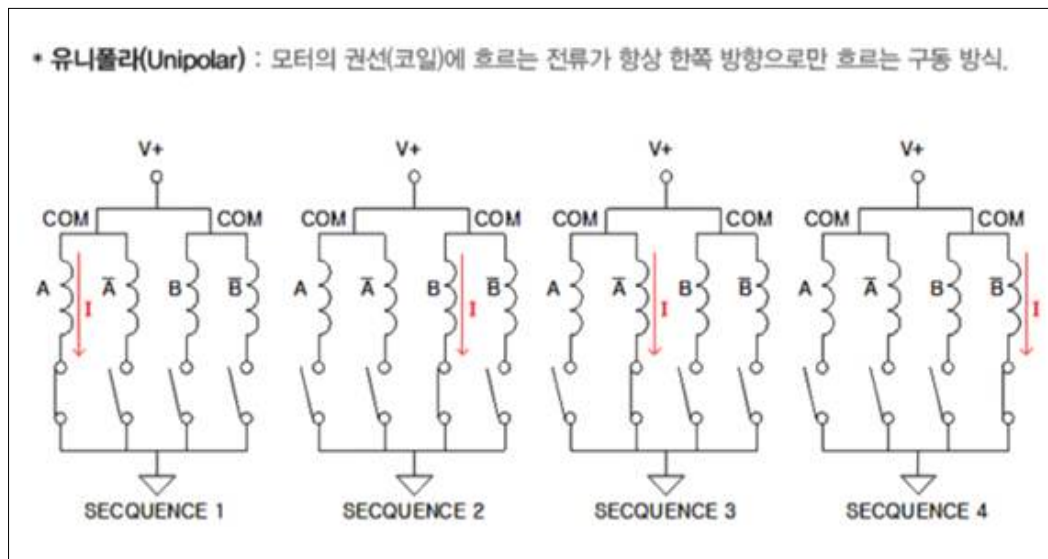
지금까지 스텝모터에 대한 기초적인 지식을 알아보았다. 스텝모터가 어떤 장점을 갖고 있고, 어떤 원리로 움직이는지 알았으니 사용한 모터에 대해 이야기 해보려 한다. 작품에 사용한 모터는 “28BYJ-48” 이다.



[그림 2-14] 28BYJ-48 스텝모터 사양

이 스텝모터의 4상 5선 유니폴라 방식의 스텝모터로 정격전압은 5V이다. 모터의 사양 표에서는 5.625도의 스텝각을 가진 것으로 표기되어 있으나, 1스텝을 1/64로 나누어 동작하기 때문에 1스텝각인 5.626도를 이동하기 위해서는 64펄스가 필요로 한다. 따라서 360도를 회전하기 위해선 36스텝각이 필요하고 이는 곧 4096펄스가 필요한 것으로 알 수 있다.

또한 유니폴라 방식이기 때문에 선이 4상임에도 불구하고 선이 5가닥 있는데, 이 선은 COM선으로 전원부에 연결해준다. 유니폴라 방식은 한 방향으로만 전류가 흐르는 방식으로 회로가 간단하고 탈조될 위험이 적다.



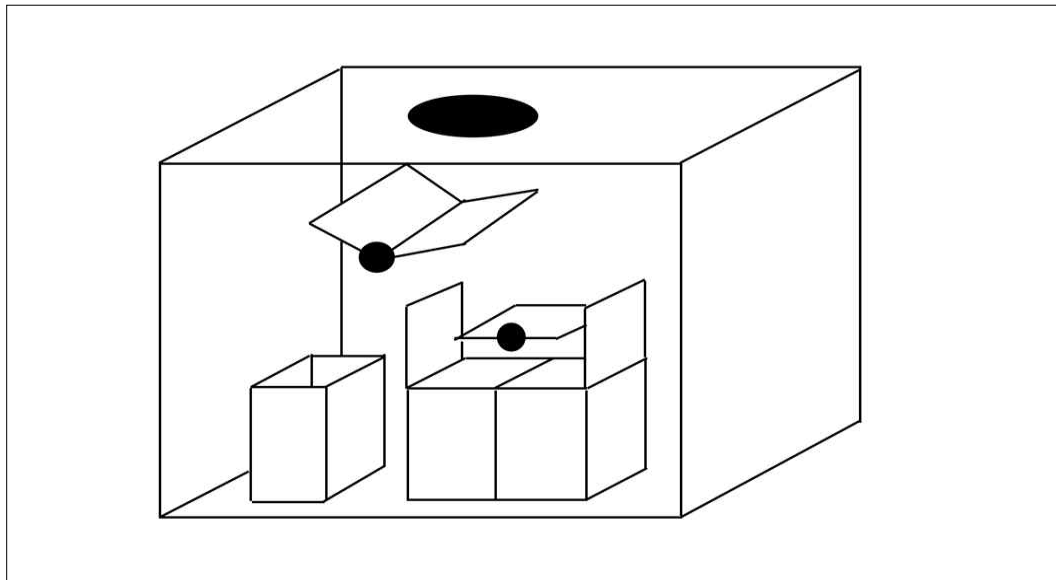
[그림 2-15] 유니폴라 구동

COM선을 제외한 나머지 선 4개는 각각의 코일에 연결되어 있으며 전자석에 전류를 흘려주는 순서가 일치해야 모터가 구동되기 때문에 브레드보드에 모터를 연결하여 순서를 찾아주는 것은 중요한 작업이므로 반드시 알아두어야 할 정보이다.

3. 제작과정

3.1 하드웨어

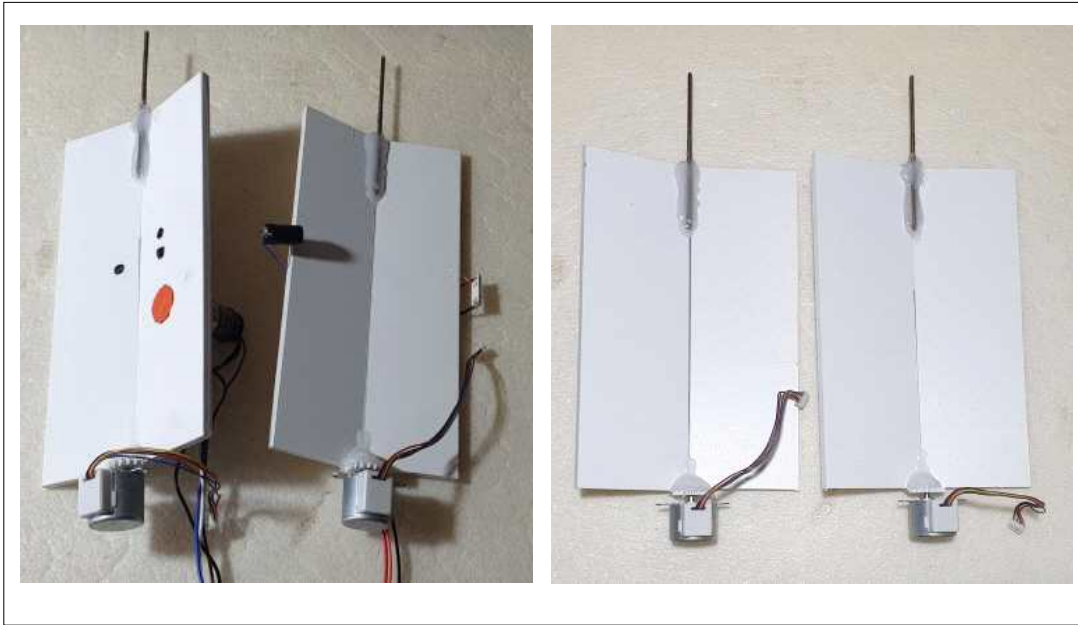
작품을 설계하기에 앞서 우선 외형을 설계하였다. 작품은 크게 “1분류”와 “2분류”로 나누었으며, 1분류에서는 금속센서를 이용하여 금속과 비금속을, 2분류에서는 플라스틱과 종이를 분류하고 1분류와 2분류에서 분류가 끝나면 스텝모터가 회전하여 쓰레기를 정해진 구역으로 낙하시키도록 작품을 구상하였다.



[그림 3-1] 스마트 분리수거 구상도

작품이 구동되는 순서는 금속(캔) - 비금속(플라스틱, 종이)의 분류가 먼저 이루어지고, 종이 - 플라스틱의 분류가 마지막으로 이루어진다. 물체를 감지하면 작동하도록 적외선 센서를, 금속인지 아닌지 구분하기 위해 근접센서를 1분류에 같이 설치하여 금속과 비금속의 분류를 하도록 설계했다. 2분류에서는 종이와 플라스틱의 분류를 하기 위해 조도센서와 발광LED를 설치했는데, LED에서 나온 빛을 조도센서를 향하도록 설치하고 쓰레기가 LED와 조도센서 사이에 떨어지도록 설계하였다.

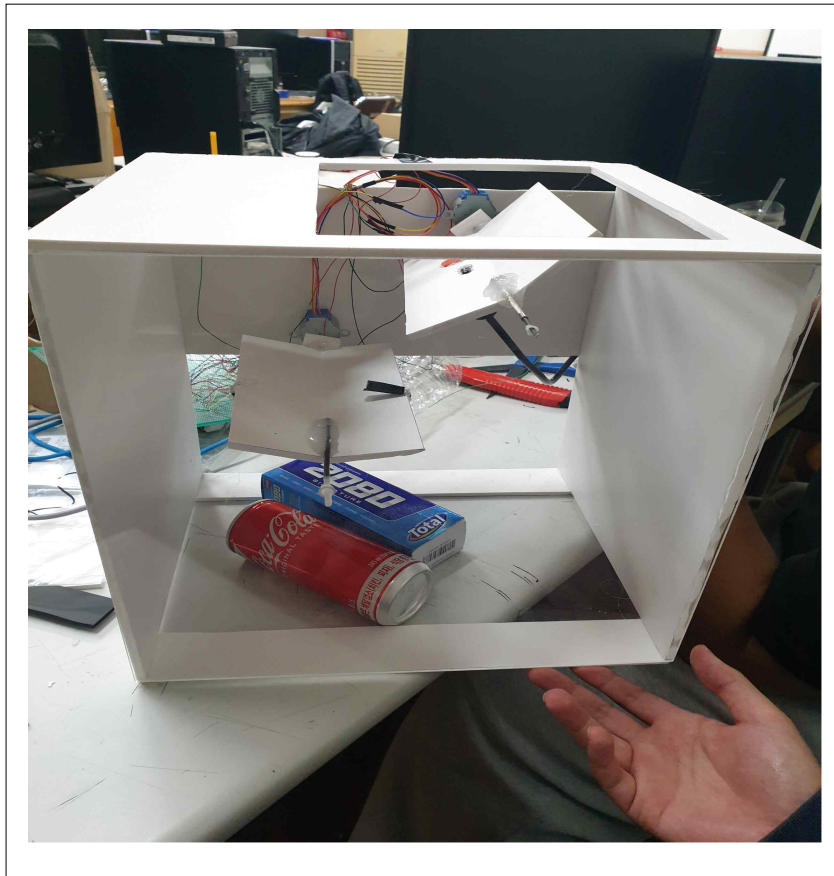
작품에서 중요한 1분류와 2분류의 받침대를 완만한 V자로 만들어서 물체가 잘 착지하도록 만들었다. 작품을 만들 때 대부분을 흰색 포맥스를 잘라서 만들었고 글루건을 이용해 접착을 했다. 모터와 받침대도 예외는 아니었는데 모터와 외관을 글루건만 이용해 접착을 하다 보니 불안해 보였고, 쓰레기가 낙하된다면 모터의 접착이 버티지 못할 것 같아서 받침대에 지지대역할을 해줄 막대를 추가로 설치했다.



[그림 3-2] 분류 받침대 사진

그리고 작품의 앞면은 쓰레기가 분류되는 과정을 보여주기 위해 투명한 아크릴을 잘라서 사용했는데, 아크릴판에 받침대가 통과할 구멍을 뚫어서 쓰레기의 낙하를 받았을 때 충격을 모터를 접착한 부분에만 부담되지 않도록 설계했다.

모터의 부착도 받침대가 닿지 않도록 크기를 측정하고 충분한 공간을 확보하여 설계하였고, 내부에서 여러 센서들의 선이 받침대의 회전시, 움직임을 방해하지 않도록 작품의 내벽에 고정하거나 외부로 나오도록 마감하였고, 시운전으로 작품이 원활하게 움직이는 것을 확인하였다.



[그림 3-3] 작품 시운전

3.2 소프트웨어

시스템이 이해할 수 있는 언어인 C언어를 입력해 동작할 수 있는 과정을 코딩이라고 한다. 스마트 분리수거의 동작을 위해 C언어를 사용하여 프로그래밍을 진행하였다. 소프트웨어 설계는 ATMEGA128의 입/출력 포트 정의 부분과 모터 제어부로 나눌 수 있다. ATMEGA128에는 7개의 포트가 내장되어 있는데 포트는 설정에 따라서 각 비트는 입력으로 쓰일 수도 있고 출력으로 쓰일 수도 있다.

입/출력 포트 정의하기 위해서 DDR 레지스터와 PORT 레지스터를 숙지해야 한다. DDR 레지스터는 I/O 기능으로 핀이 전압을 주거나(OUTPUT) 받는(INPUT) 핀으로 사용할 수 있는데 DDR은 I/O를 선언해주는 레지스터다. OUTPUT은 자신이 원하는 핀에서 5V 전압을 출력할 것인가, 출력하지 않을 것인가 선택할 수 있는 상태이며 DDR 셋팅을 '0' 으로 하면 출력 선언을 하지 않는 것이고 '1' 로 하면 출력 선언을 하는 것이다. INPUT은 자신이 원하는 핀에서 전압을 받을 것인가, 받지 않을 것인가 선택할 수 있는 상태이며 DDR 셋팅을 '0' 으로 하면 전압을 받는다고 선언하는 것이고 '1' 로 하면 전압을 받지 않는다고 선언하는 것이다.

PORT 레지스터는 DDR 레지스터를 이용하여 OUTPUT 선언된 포트를 제어하는 레지스터이며 OUTPUT으로 선언된 포트만 PIN을 사용할 수 있다. PORT를 사용하면 OUTPUT으로 선언한 포트에서 5V를 출력할 것인지, 출력하지 않을 것인지 제어할 수 있다.

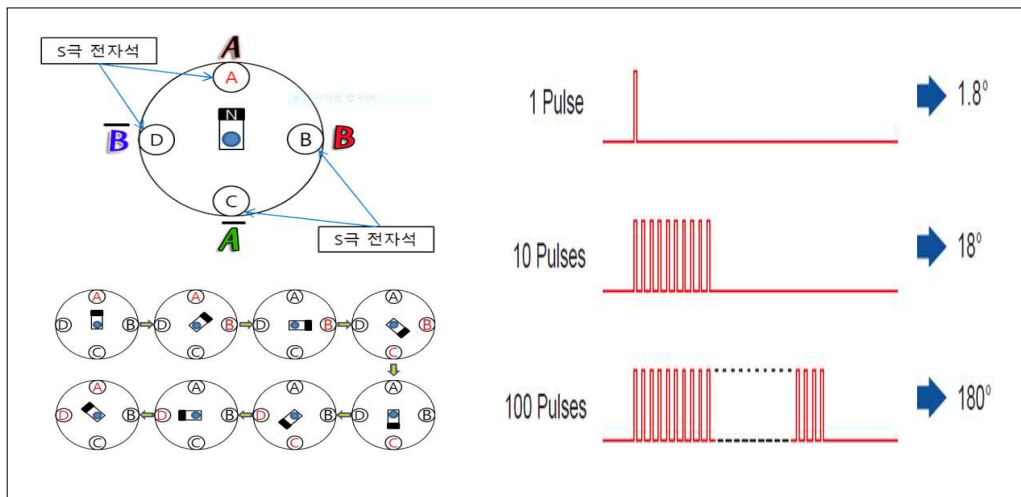
PIN은 DDR 레지스터를 이용하여 INPUT 선언된 포트를 제어하는 레지스터로서 INPUT으로 선언된 포트만 PIN으로 사용할 수 있다. PIN을 사용하면 INPUT으로 선언한 포트에서 전압을 받을 것인지, 받지 않을 것인지 제어할 수 있다.

ATMEGA128의 입/출력 포트는 [표 3-1] 와 같이 포트 E에 적외선 센서, 근접 센서, 조도 센서를 입력 포트에 사용하였고, 포트 B와 D는 스텝모터 출력 포트에 사용하였으며 정의를 하고 설계하였다.

코딩 설계	내용
<pre>int main(void) { DDRB = 0XFF; DDRD = 0XFF; DDRE = 0X00;</pre>	<p>B포트 초기화(상측모터 연결됨)</p> <p>D포트 초기화(하측모터 연결됨)</p> <p>각종 센서 연결됨.</p>

[표 3-1] 입/출력 포트 정의 코딩 설계

센서에서 출력된 신호로 스텝모터를 제어하기 위한 코딩은 [표 3-2]와 같이 설계되었다. 모터의 회전 각은 [그림 3-1] S극 전자석에 전류가 흐르면 회전축과 연결된 N극 전자석이 당겨지면서 회전하게 되는데 HIGH 신호를 받아 S극 전자석이 한 주기 돌면 스텝 각은 5.625° 이다. 모터가 한 바퀴 돌기 위한 HIGH 펄스는 $(360/5.625) * 64 = 4096$ 이다.



[그림 3-4] 스텝모터 전자석과 스텝 각

작품에서 모터의 회전을 통해 분류 판이 돌아가는 것을 목표로 사용하였다. 분류 판의 기울기는 외형에 따라 달라짐으로 처음에 생각했던 HIGH 펄스는 오차가 생겼기에 직접 조정하면서 맞춰갔다. 적당한 기울기 각도가 나온 것은 120번의 HIGH 펄스였으며 회전속도를 결정하기 위해 Delay 함수를 사용하였다.

코딩 설계	내용
//---- 아래 모터 void MOTOR1RR(void) {register unsigned int i; {for(i = 0; i<120; i++) {PORTD= 0B10001000; Delay_ms(5); PORTD= 0B01000100; Delay_ms(5); PORTD= 0B00100010; Delay_ms(5); PORTD= 0B00010001; Delay_ms(5);}	120회 반복하여 기울기 각도를 결정 Delay 함수를 사용해서 회전속도 조절 - 아래 모터 - MOTOR1RR은 모터가 오른쪽으로 일정 기울기 각도 회전 MOTOR1LL은 모터가 왼쪽으로 일정 기울기 각도 회전

<pre> } } void MOTOR1LL(void) {register unsigned int i; {for(i = 0; i<120; i++) {PORTD= 0B00010001; Delay_ms(5); PORTD= 0B00100010; Delay_ms(5); PORTD= 0B01000100; Delay_ms(5); PORTD= 0B10001000; Delay_ms(5);} } } //--- 위 모터 void MOTOR2RR(void) {register unsigned int i; {for(i=0; i<120; i++) {PORTB= 0B10001000; Delay_ms(5); PORTB= 0B01000100; Delay_ms(5); PORTB= 0B00100010; Delay_ms(5); PORTB= 0B00010001; Delay_ms(5);} PORTB= 0B00000000; } } void MOTOR2LL(void) {register unsigned int i; {for(i=0; i<120; i++) {PORTB= 0B00010001; Delay_ms(5); PORTB= 0B00100010; Delay_ms(5); PORTB= 0B01000100; Delay_ms(5); </pre>	<p>- 위 모터 -</p> <p>MOTOR2RR은 모터가 오른쪽으로 일정 기울기 각도 회전</p> <p>MOTOR2LL은 모터가 왼쪽으로 일정 기울기 각도 회전</p>
---	--

<pre> PORTB= 0B10001000; Delay_ms(5);} PORTB= 0B00000000; } } </pre>	
---	--

[표 3-2] 스텝모터 기울기와 회전속도 결정

1차 분류에서 쓰이는 센서는 적외선 센서와 근접 센서이다. 적외선 센서는 다른 센서와 다르게 항상 HIGH(1) 신호를 출력하고 있다. 물체가 감지되었을 때 LOW(0) 신호를 출력하기에 주의하고 코딩했다. 근접 센서는 자성 물체 (금속)이 다가오면 HIGH(1) 신호를 출력하는데 비자성 금속인 알루미늄도 검출이 가능하지만 검출 거리가 짧아진다는 단점이 있다. 비금속 물질을 감지했을 때는 LOW(0) 신호를 출력한다. [표 3-3]과 같이 금속을 감지하면 적외선 센서와 근접 센서가 활성화되고 모터가 오른쪽으로 돌았다가 다시 왼쪽으로 돌아서 원점으로 돌아오게 설계했다. 비금속일 경우 적외선 센서만 감지했기에 모터가 왼쪽으로 돌았다가 다시 원점으로 돌아온다.

2차 분류에서 쓰이는 센서는 조도 센서이다. 조도 센서의 성질은 빛을 받으면 저항값이 낮아지고 빛을 받지 못하면 저항값이 커진다. 이 성질을 이용하여 전압분배 법칙을 사용하여 조도 센서에서 ATMEGA128로 출력 신호를 보내준다. $V = (R / (CdS + R)) * 5V$ 의 전압분배 법칙을 사용해 빛을 받아 저항이 작아지면 5V에 가깝게 출력이 되기에 HIGH(1) 신호를 출력하고 빛을 받지 못하면 저항값이 커지기에 0V에 가깝게 출력되기에 LOW(0) 신호를 출력한다. 작품에서 사용하는 비금속 쓰레기는 플라스틱과 종이이기에 플라스틱은 발광 LED를 통해 빛이 투과하여 조도 센서에 감지가 되기에 HIGH(1) 신호를 출력하고 종이는 빛을 막기에 조도 센서는 빛을 감지하지 못한다. 그렇기에 LOW(0) 신호를 출력한다. 조도 센서가 HIGH(1) 신호를 출력했을 때는 모터가 왼쪽으로 돌았다가 다시 원점으로 돌아오도록, LOW(0) 신호를 출력했을 때는 모터가 오른쪽으로 돌았다가 다시 원점으로 돌아오도록 설계하였다.

코딩 설계	내용
<pre> //--- 상측 분리 if(!(PINE & 0B00100000)) { Delay_ms(2000); if((PINE & 0B00010000)) {MOTOR2RR(); MOTOR2LL(); } } </pre>	<p>- 상측 -</p> <p>적외선 감지</p> <p>금속 감지</p> <p>모터 - 우측 회전, 원위치</p> <p>금속 감지를 하지 못했을 경우</p> <p>모터 - 좌측 회전, 원위치</p>

<pre> else {MOTOR2LL(); MOTOR2RR(); //--- 하측 분리 if((PINE & 0B00001000)) {MOTOR1LL(); MOTOR1RR(); } else {MOTOR1RR(); MOTOR1LL(); } } } </pre>	<p>- 하측 -</p> <p>조도 센서 빛 감지</p> <p>모터 - 좌측 회전, 원위치</p> <p>조도 센서 빛 감지하지 못했을 경우</p> <p>모터 - 우측 회전, 원위치</p>
--	---

[표 3-3] 센서 감지를 통한 모터 제어

4. 결론

4.1 시스템 제작

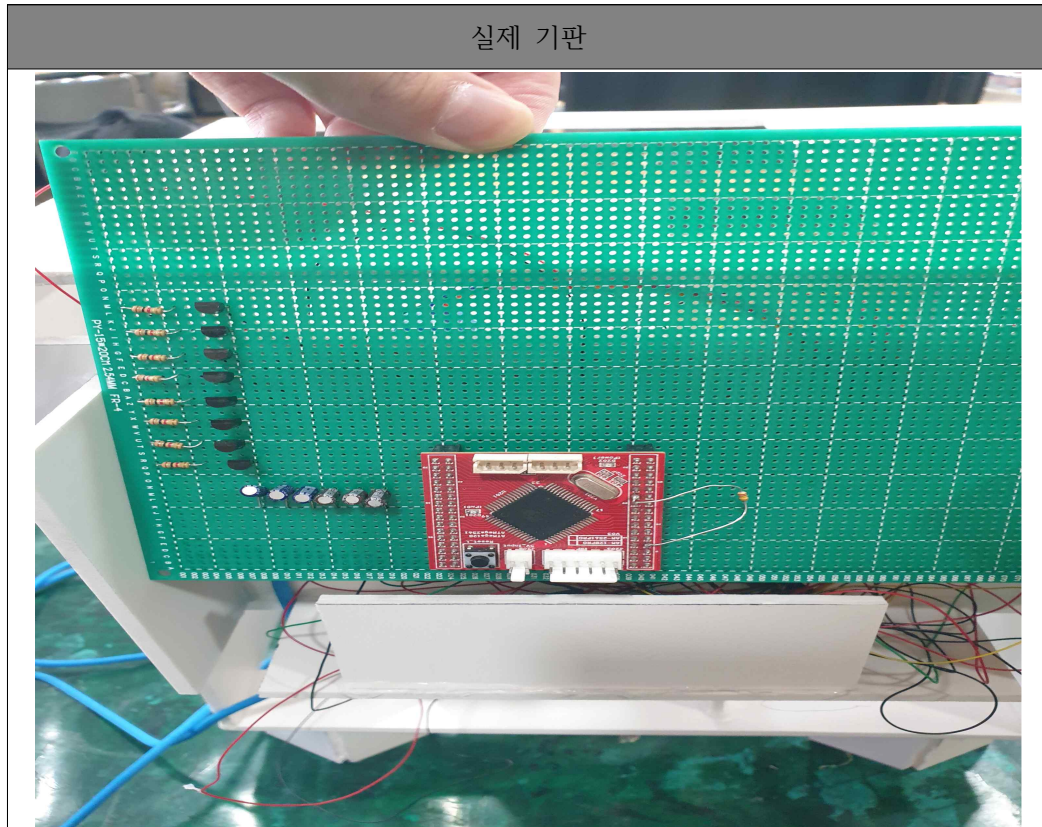
4.2 최종동작

4.3 결론

4.4 부록


4.1 시스템 제작

ATMEGA128을 중심으로 외부전원 어댑터 5V를 사용하였고, 적외선 센서와 근접 센서, 조도 센서와 스텝 모터를 연동하여 사용하였다. ATMEGA128의 잡음과 스텝 모터를 구동하기 위해 커패시터와 트랜지스터를 사용하였다. 트랜지스터는 ATMEGA128의 출력신호를 받아 스위칭 작용을 한다. [그림 4-1]과 같이 트랜지스터, 커패시터, 저항을 이용하여 기판을 제작하였다.



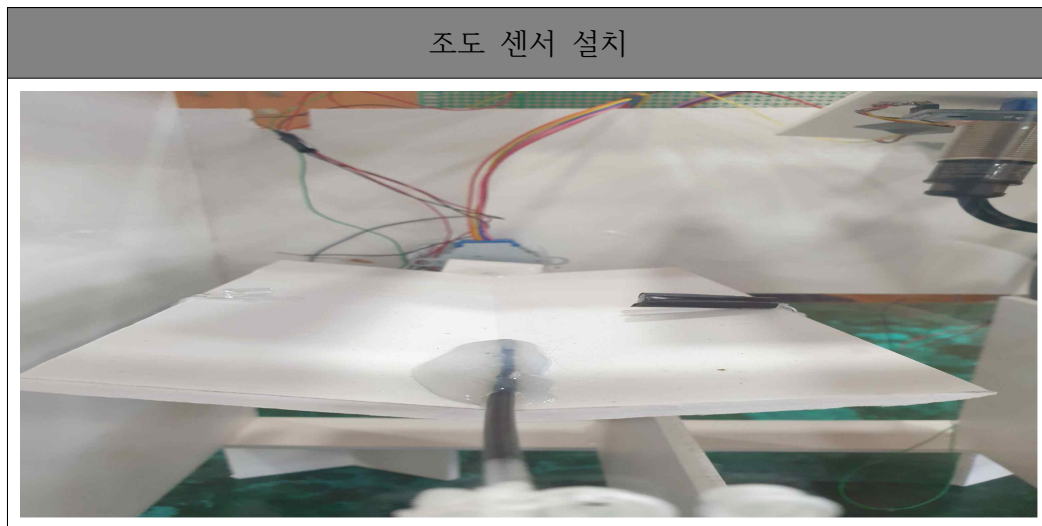
[그림 4-1] 최종 연결 기판

1차 분류 판에 적외선 센서와 근접 센서를 사용하게 되었다. 적외선 센서는 쓰레기가 투입됐음을 감지해야하기 때문에 감지 거리를 2cm로 조정해냈다. 감지 거리를 작게 한 이유는 근접 센서의 감지 거리가 8mm이기에 최대한 인식이 같이 되도록 설정하였다. 적외선 센서의 감지 거리를 최대로 줄여도 근접 센서와 차이가 있기에 [그림 4-2]와 같이 적외선 센서 코딩에 딜레이를 줘서 근접 센서와 같이 인식할 수 있도록 하였다.

1차 분류 센서	딜레이 코딩
	<pre> if(!(PINE & 0B00100000)) 적외선 감지 { Delay_ms(2000); if((PINE & 0B00010000)) 금속 { MOTOR2RR(); MOTOR2LL(); } else 금속 아니면(펄트, 종이) {MOTOR2LL(); MOTOR2RR(); </pre>

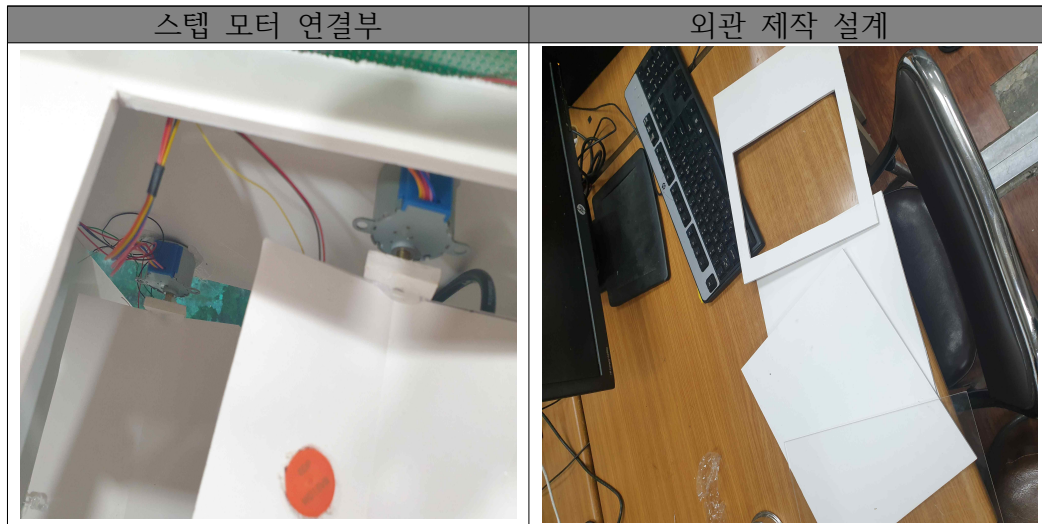
[그림 4-2] 1차 분류 판 설명

2차 분류 판에는 조도 센서를 사용하게 되었다. 조도 센서를 사용할 때 주위의 빛에 영향을 많이 받아서 [그림 4-3]과 같이 절연 테이프로 감아, 빛을 모아 받을 수 있도록 제작하였다. 전압분배법칙을 사용하여 빛을 받지 못했을 때 민감하게 반응할 수 있도록 설계하였다.



[그림 4-3] 2차 분류 판 센서

스텝 모터는 [그림 4-4]와 같이 벽 쪽에 글루건으로 붙이고 포맥스로 만들어진 분류 판을 글루건으로 스텝 모터와 연결하였다. 쓰레기가 떨어질 때 충격으로 인해 모터가 떨어질 것을 방지하기 위하여 전방 아크릴에 구멍을 뚫어 모터의 지지대를 꽂아 주었다. 외관은 포맥스, 아크릴을 이용했으며 글루건으로 작업을 진행했다.



[그림 4-4] 스텝 모터와 외관 제작

4.2 최종 동작

스마트 분리수거는 적외선 센서로 물체가 투입되는 것을 감지한 후 근접 센서로 금속/비금속으로 나누게 된다. 적외선 센서는 물체가 투입됐을 때 LOW 신호를 ATMEGA128에게 송신하게 된다. 근접 센서는 금속 재질의 물체가 다가왔을 때 HIGH 신호를 송신하게 되는데, 적외선 센서와 근접 센서가 모두 감지하였을 때는 금속 물체이기에 모터가 오른쪽으로 돌아 분류하게 되고, 적외선 센서만 감지되었을 시에는 금속이 아니기에 모터가 왼쪽으로 돌아 비금속 분류작업을 하기 위해 2차 분류로 넘어간다.

2차 분류작업은 조도 센서로 플라스틱과 종이를 구별하는 작업을 한다. 조도 센서 맞은편에 발광 다이오드를 설치하여 쓰레기에 빛을 쏘주어서 빛을 통과하는 것을 조도 센서가 감지하는 것이다. 플라스틱일 경우 빛이 투과되어 조도 센서로 그대로 들어가서 HIGH 신호를 송신하게 되고 모터는 왼쪽으로 돌게 된다. 종이일 경우는 빛을 막아 조도 센서로 들어가는 빛이 없기에 LOW 신호를 송신하게 되고 모터는 오른쪽으로 돌게 된다.

[그림 4-5]와 같이 외관 제작 이후에 캔, 플라스틱, 종이의 분류가 오차 없이 최종 동작하는 것을 확인하였다.



[그림 4-5] 스마트 분리수거 최종 구현

4.3 결론

작품을 만들면서 다양한 쓰레기들을 분류하고 싶었지만 제한된 지원금과 시간으로 분류할 수 있는 쓰레기들이 많지 않다는 것에 대한 한계를 느꼈다. 또한 분류할 수 있는 쓰레기의 종류가 다양하지 않아 일상생활에서 사용할 때 편하지 않을 것이고 쓰레기를 넣을 때 하나씩 넣어야 한다는 점이 아쉬운 점이다. 이를 개선하기 위해선 부피를 더 키워 기능을 추가하고 다수의 쓰레기를 넣었을 때 인식할 수 있는 방법을 찾아야 한다. 또한 투입구에 별도의 문을 달아 낙엽이나 빗물 등 야외에서 발생할 수 있는 다양한 상황에서 센서의 오작동을 최소화해야 한다. 이렇게 아쉬움도 많고 어려움도 있었지만, 팀 활동으로 센서와 모터를 우리가 의도한 방향으로 제어하기 위해서 자료를 찾고 공부를 하면서 어떤 수업들보다 많은 공부를 하였다. 또한 직접 부품들을 제어하면서 얻었던 즐거움과 성취감은 지금까지의 수업에서는 느낄 수 없었던 감정이었다.

4.4 부록

코딩설계

```
#include <avr/io.h>                //헤더파일 포함

void Delay_us(unsigned char time_us) /* time delay for us */
{ register unsigned char i;

  for(i = 0; i < time_us; i++)      // 4 cycle +
    { asm volatile(" PUSH   R0 ");  // 2 cycle +
      asm volatile(" POP    R0 ");   // 2 cycle +   = 8 cycle = 1 us for
8MHz

      asm volatile(" PUSH   R0 ");   // 2 cycle +
-
      asm volatile(" POP    R0 ");   // 2 cycle +
-
    }
```

```

asm volatile(" PUSH  R0 ");    // 2 cycle +
-
asm volatile(" POP   R0 ");    // 2 cycle = 16 cycle = 1 us for 16MHz
-
//-----
}
}

```

```

void Delay_ms(unsigned int time_ms)    /* time delay for ms */
{ register unsigned int i;

```

```

    for(i = 0; i < time_ms; i++)
    { Delay_us(250);
      Delay_us(250);
      Delay_us(250);
      Delay_us(250); }
}

```

```

//----- 아래 모터 -----

```

```

void MOTOR1RR(void)    //아래 모터 우측 기울이기
{register unsigned int i;
  {for(i = 0; i<120; i++)    //120회 반복해서 기울임 각도만들어짐.
    {PORTD= 0B10001000; Delay_ms(5); //회전속도 결정
      PORTD= 0B01000100; Delay_ms(5);
      PORTD= 0B00100010; Delay_ms(5);
      PORTD= 0B00010001; Delay_ms(5);}
    }
}

```

```

void MOTOR1LL(void)    //아래모터 좌측 기울이기
{register unsigned int i;
  {for(i = 0; i<120; i++)
    {PORTD= 0B00010001; Delay_ms(5);
      PORTD= 0B00100010; Delay_ms(5);
      PORTD= 0B01000100; Delay_ms(5);
      PORTD= 0B10001000; Delay_ms(5);}
    }
}
}

```

```

//----- 위 모터 -----
void MOTOR2RR(void)      //상측 모터 우측 기울이기
{register unsigned int i;
  {for(i=0; i<120; i++)
    {PORTB= 0B10001000; Delay_ms(5);
      PORTB= 0B01000100; Delay_ms(5);
      PORTB= 0B00100010; Delay_ms(5);
      PORTB= 0B00010001; Delay_ms(5);}
    PORTB= 0B00000000;
  }
}

void MOTOR2LL(void)      //상측모터 좌측 기울이기
{register unsigned int i;
  {for(i=0; i<120; i++)
    {PORTB= 0B00010001; Delay_ms(5);
      PORTB= 0B00100010; Delay_ms(5);
      PORTB= 0B01000100; Delay_ms(5);
      PORTB= 0B10001000; Delay_ms(5);}
    PORTB= 0B00000000;
  }
}

int main(void)
{ DDRB = 0XFF;      //B포트 초기화(상측모터 연결됨)
  DDRD = 0XFF;      //D포트 초기화(하측모터 연결됨)
  DDRE = 0X00;      //각종 센서 연결됨.

  while(1){        //무한 반복 하기를.....
//----- 상측 분리 -----
    if(!(PINE & 0B00100000)) //상측 적외선 감지 되면...
    {
      Delay_ms(2000);
      if((PINE & 0B00010000))//그리고 금속 감지되면(캔)
      {
        MOTOR2RR();          //우측 떨어트리고
        MOTOR2LL();          //원위치
      }
    }
  }
}

```

```

    }
else                                     //금속 아니면(펄트, 종이)
{MOTOR2LL();                             //좌측 떨어트리고
  MOTOR2RR();                             //원위치

//----- 하측 분리 -----
  if((PINE & 0B00001000))                //빛이 차단되면(종이)
    {MOTOR1LL();                          //우측(중앙) 떨어트리고
      MOTOR1RR();                          //원위치
    }
  else                                    //빛 차단 아니면(투명펄트)
    {MOTOR1RR();                          //좌측 떨어트리고
      MOTOR1LL();                          //원위치
    }
  }
}

}
}

```

참고문헌

- [1] 윤왕철, “AVR ATmega 128”, 북두출판사, 2014.08.05.
- [2] 허경용, “ATmega128로 배우는 마이크로컨트롤러 프로그래밍” , Jpub, 2016.11.25.
- [3] 김상훈, “모터제어” , 북두출판사, 2022.01.05.

