12조

스마트 수족관

15420057 남상호 15421034 최영민

목차

1 서론

연구의 동기

인공지능(AI)

반려동물

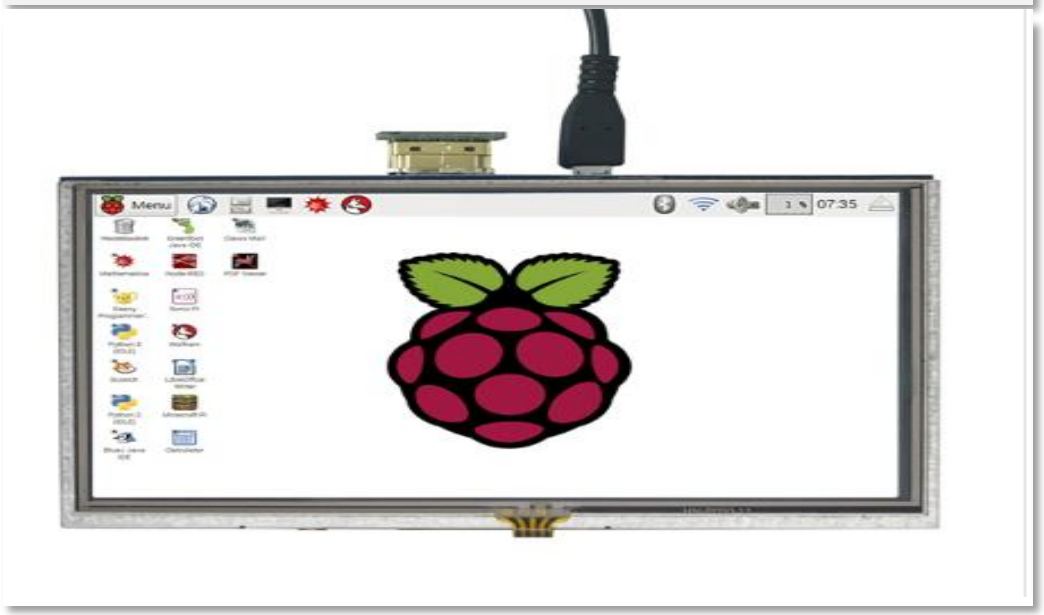
학교에서 서울 전자전으로 견학을 간 적이 있습니다. 전자전을 견학하면서 여러 가지 작품들과 기업에서 새롭게 선보이는 제품들이 있었습니다. 그것들을 보면서 느꼈던 것이 지금 현재 생활에 너무 적응하고 생활한 탓에 기술이 이정도로 발전 했었나 라는 놀라움이 있었습니다. 그래서 집으로 돌아와 현재 진행되고 있는 제 4차 산업혁명에 대해 검색을 해 보았습니다. 제 4차 산업혁명은 빅데이터, 인공지능, 사물인터넷 등을 이용해 여러 분야에 활용되어지고 있었습니다. 그 중 센서 데이터 전송 기술을 사용해 평소에 관심이 있었던 반려동물 분야에 접합시켜 졸업 작품을 만들어 보자는 목표가 생겼고 예전에 열대어를 기르면서 필요하다고 느꼈던 부분이나 불편했던 점을 개선해서 졸업작품을 제작하게 되었습니다.

1 서론

연구의 목적

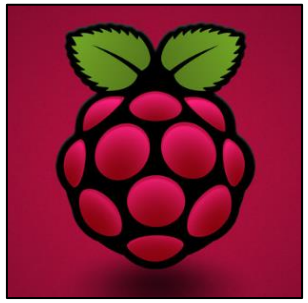
분리되어 있는 동작 버튼

터치 스크린

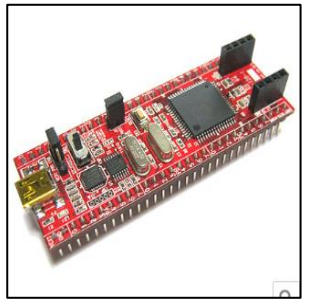


2 연구방법

동작 흐름도



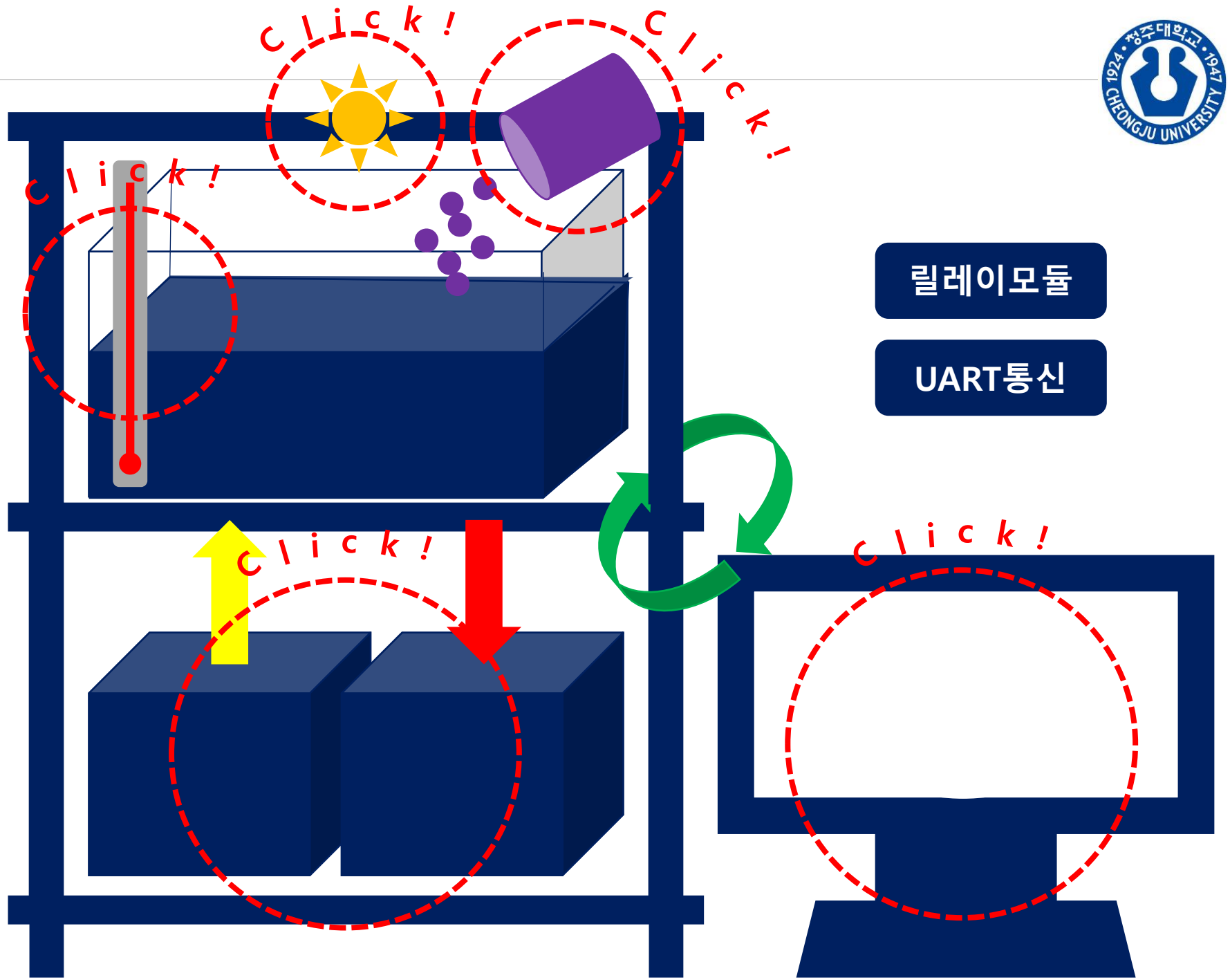
UART



온 도 설 정	온 도 센 서	→	히 터 동 작	
LED ON/OFF		→	LED ON/OFF	
급 수		→	수 위 센 서	모 터
배 수		→	수 위 센 서	모 터
FEED		→	사 료 공 급	

2 연구방법

부품과 소프트웨어



제품 모형도

2 연구방법

부품과 소프트웨어

마니 LED 플레시블 3628 LED바 흰띠 12V 6M롤

1. 같이 사용한 부품
- 4CH 릴레이 모듈
- DC 12V 전원

2. 사용 소프트웨어
- CodeVision AVR
- Raspbian(Python3)

2 연구방법

부품과 소프트웨어

DS18B20 방수형 온도센서 모듈 / 히터봉

1. 같이 사용한 부품

- Raspberry Pi
- ATmega128

2. 사용 소프트웨어

- Raspbian(Python3)
- Codevision AVR

2 연구방법

부품과 소프트웨어

사료 공급 장치

1. 같이 사용한 부품
- ATmega128
- 4채널 릴레이 모듈

2. 사용 소프트웨어
- CodeVision AVR

2 연구방법

부품과 소프트웨어

사료 공급 장치





제작 과정

2 연구방법

부품과 소프트웨어

수중펌프모터 [SZH-GNP155], 수위센서

1. 같이 사용한 부품

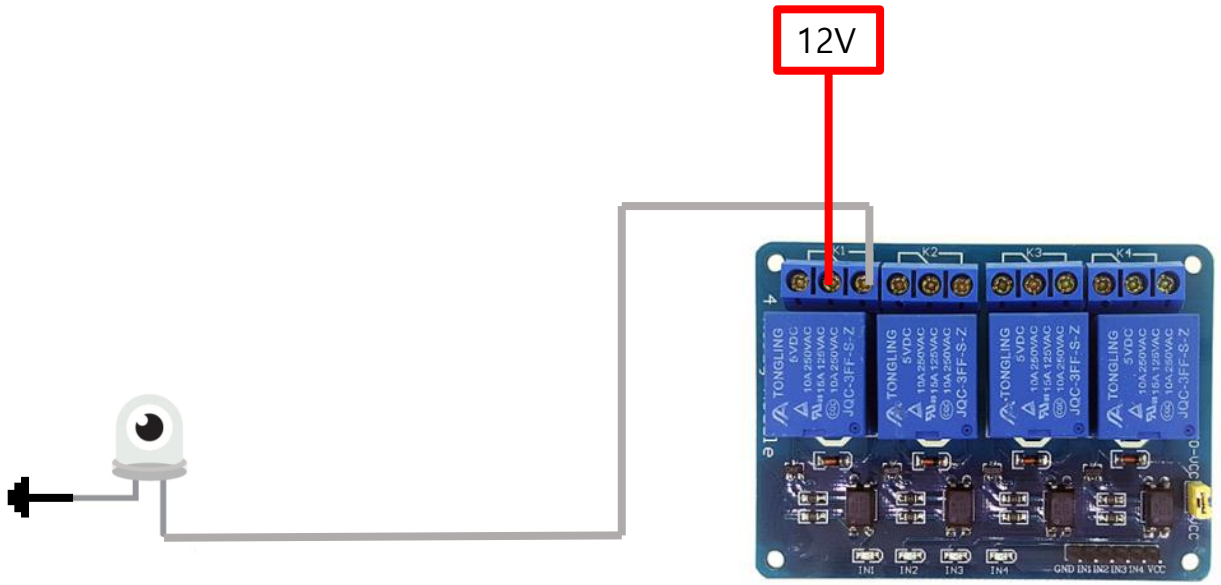
- ATmega128
- 4채널 릴레이 모듈

2. 사용 소프트웨어

- CodeVision AVR

2 연구방법

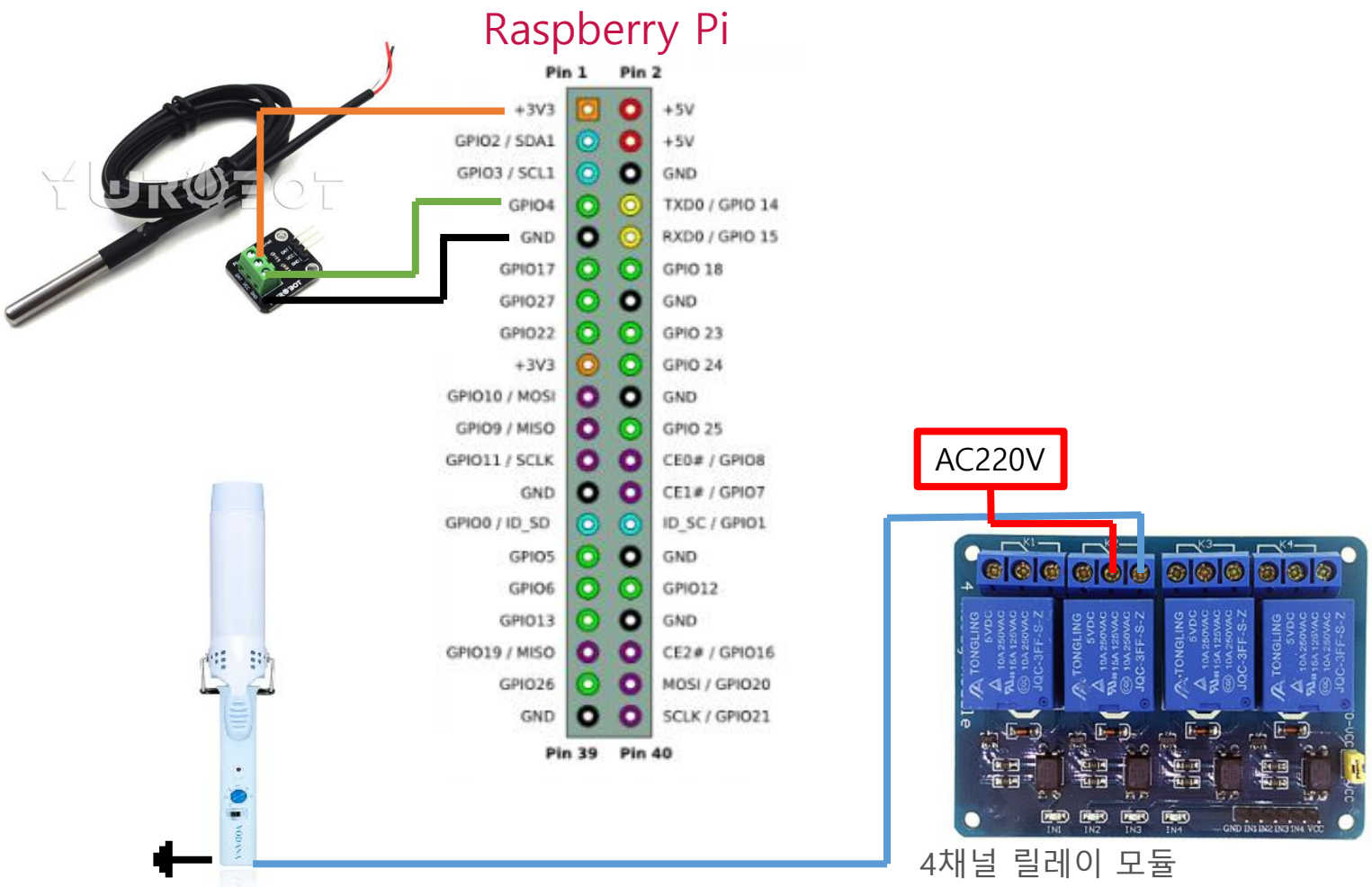
회로도



4채널 릴레이 모듈

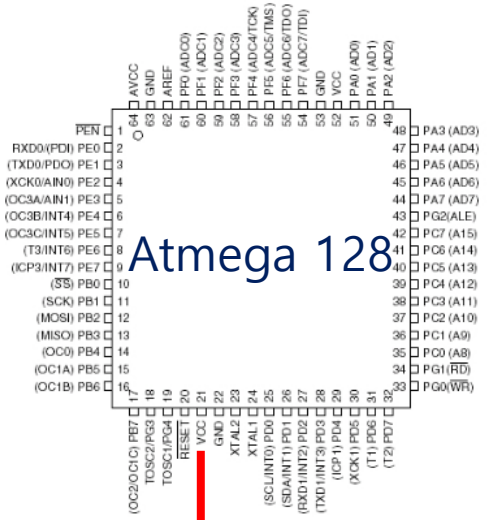
2 연구방법

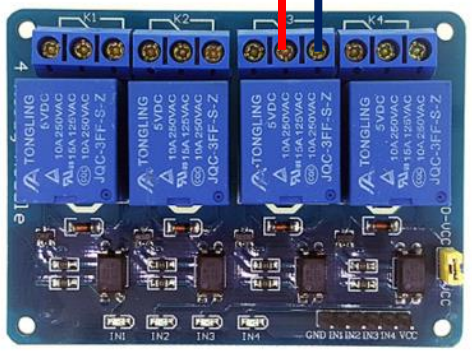
회로도


Raspberry Pi
Pin 1 Pin 2
+3V3 +5V
GPIO2 / SDA1 +5V
GPIO3 / SCL1 GND
GPIO4 TXD0 / GPIO 14
GND RXD0 / GPIO 15
GPIO17 GPIO 18
GPIO27 GND
GPIO22 GPIO 23
+3V3 GPIO 24
GPIO10 / MOSI GND
GPIO9 / MISO GPIO 25
GPIO11 / SCLK CE0# / GPIO8
GND CE1# / GPIO7
GPIO0 / ID_SD ID_SC / GPIO1
GPIO5 GND
GPIO6 GPIO12
GPIO13 GND
GPIO19 / MISO CE2# / GPIO16
GPIO26 MOSI / GPIO20
GND SCLK / GPIO21
Pin 39 Pin 40
AC220V
4채널 릴레이 모듈

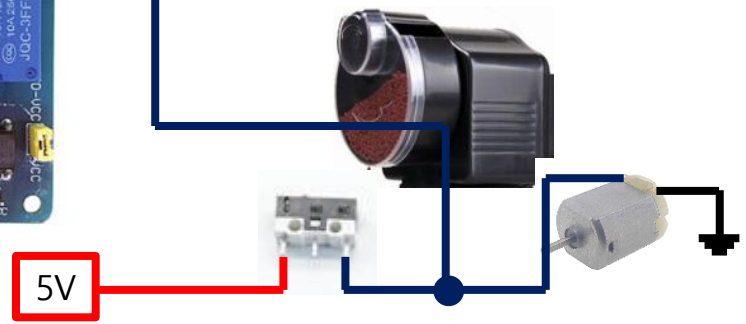
2 연구방법

회로도





4채널 릴레이 모듈



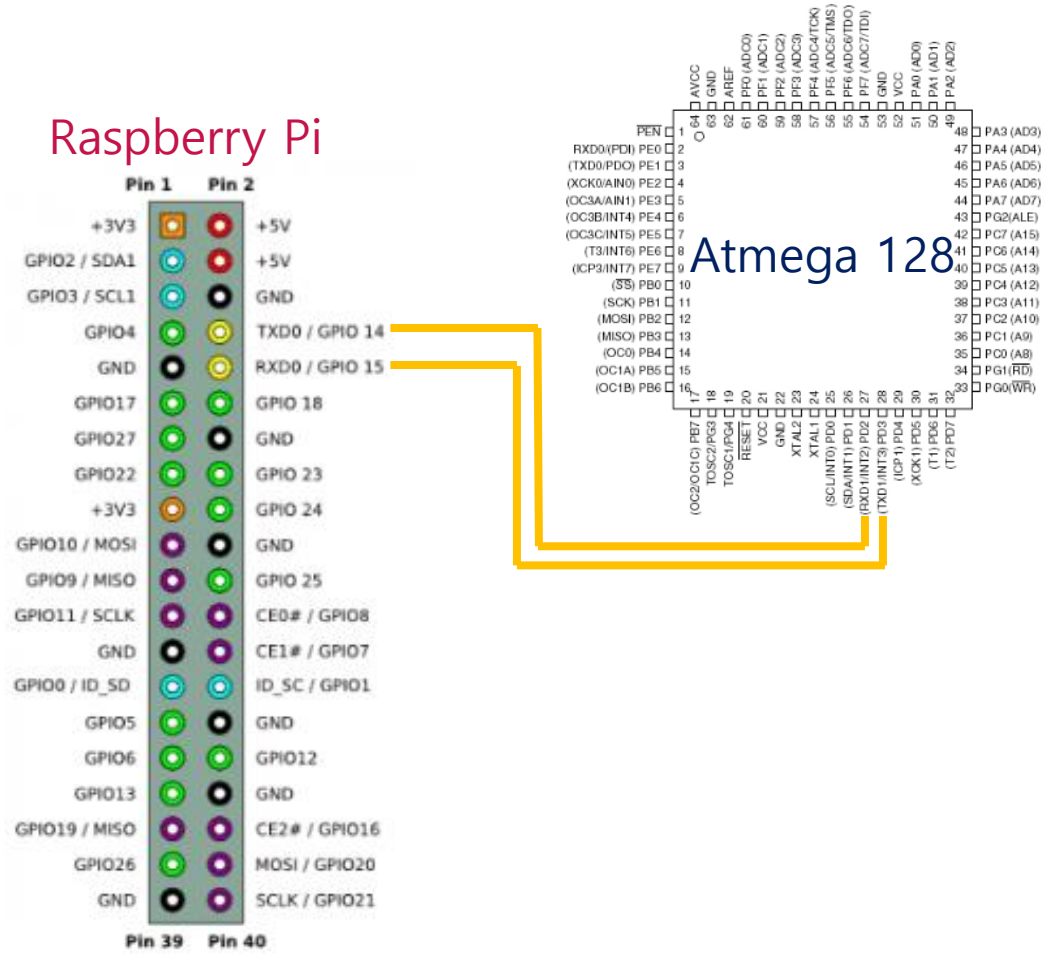
2 연구방법

회로도

동작신호
PORTA.1=LED
PORTA.2=히터
PORTA.3=사료공급장치
PORTA.4=급수펌프

2 연구방법

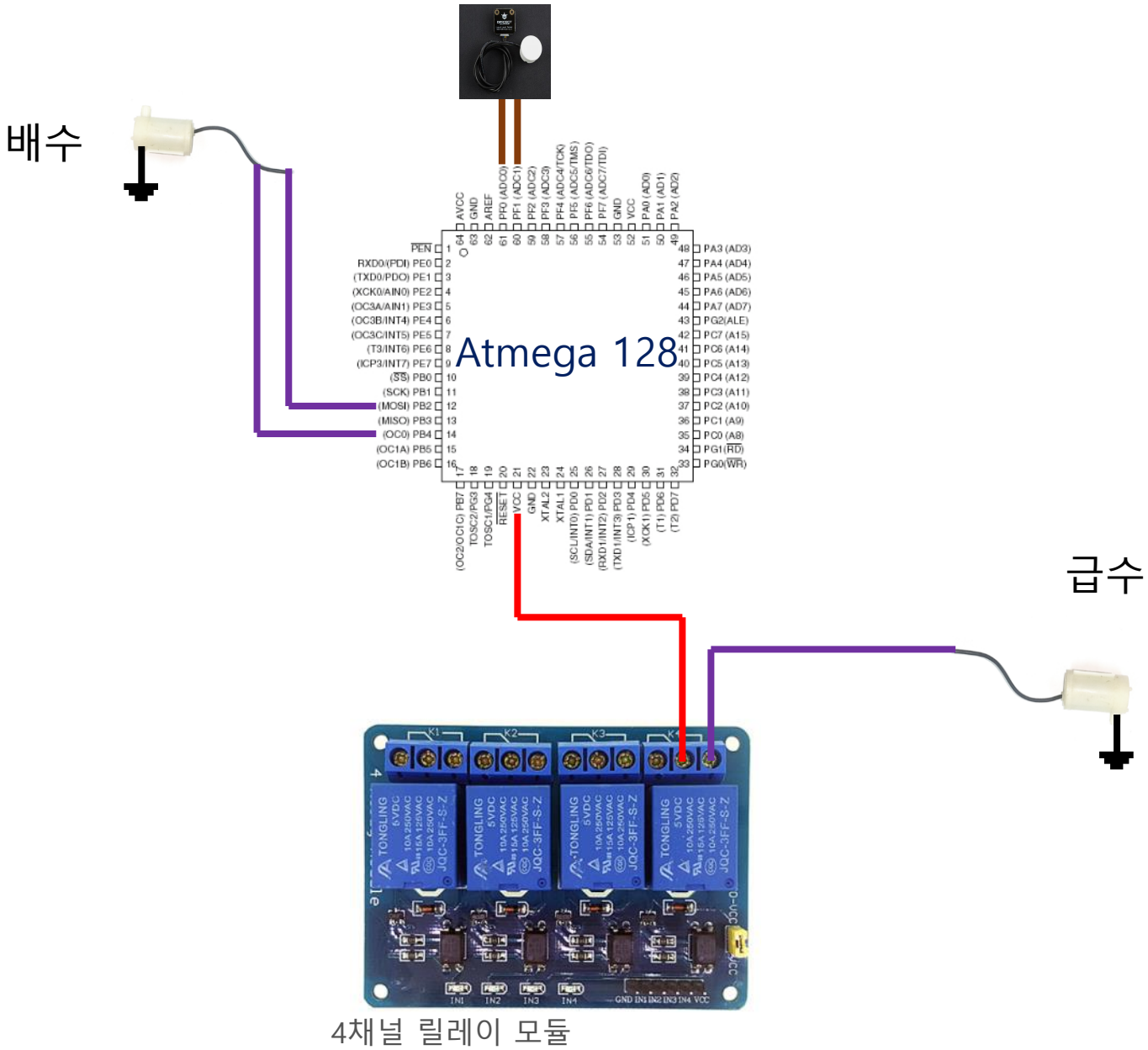
회로도



UART 통신
A = 급수
B = 배수
C = 사료 공급 장치
D = LED ON
E = LED OFF
F = 히터 ON
G = 히터 OFF

2 연구방법

회로도


배수
Atmega 128
PEN
RXD0/(PDI) PE0
(TXD0/PDO) PE1
(XCK0/AIN0) PE2
(OC3A/AIN1) PE3
(OC3B/INT4) PE4
(OC3C/INT5) PE5
(T3/INT6) PE6
(ICP3/INT7) PE7
(SS) PB0
(SCK) PB1
(MOSI) PB2
(MISO) PB3
(OC0) PB4
(OC1A) PB5
(OC1B) PB6
PA3 (AD3)
PA4 (AD4)
PA5 (AD5)
PA6 (AD6)
PA7 (AD7)
PG2(ALE)
PC7 (A15)
PC6 (A14)
PC5 (A13)
PC4 (A12)
PC3 (A11)
PC2 (A10)
PC1 (A9)
PC0 (A8)
PG1(RD)
PG0(WR)
RESET
VCC
GND
XTAL2
XTAL1
급수
4채널 릴레이 모듈

2 연구방법

부품과 소프트웨어

라즈베리파이 5인치 800x480 HDML LCD 터치스크린 모니터

1. 같이 사용한 부품

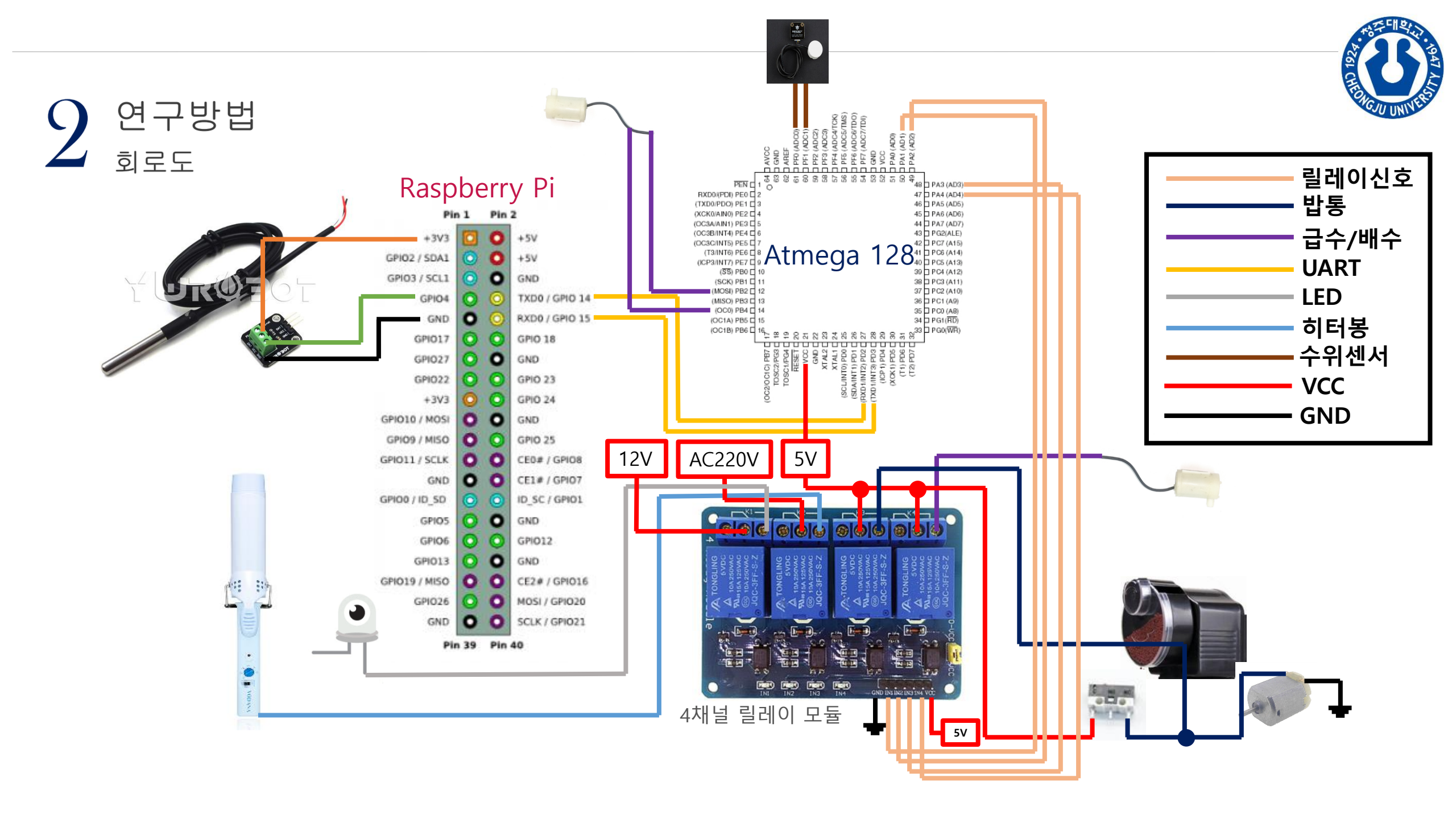
- Raspberry Pi
- HDMI Cable
- 전원

2. 사용 소프트웨어

- Raspbian

2 연구방법

회로도


Raspberry Pi
Pin 1 Pin 2
+3V3 +5V
GPIO2 / SDA1 +5V
GPIO3 / SCL1 GND
GPIO4 TXD0 / GPIO 14
GND RXD0 / GPIO 15
GPIO17 GPIO 18
GPIO27 GND
GPIO22 GPIO 23
+3V3 GPIO 24
GPIO10 / MOSI GND
GPIO9 / MISO GPIO 25
GPIO11 / SCLK CE0# / GPIO8
GND CE1# / GPIO7
GPIO0 / ID_SD ID_SC / GPIO1
GPIO5 GND
GPIO6 GPIO12
GPIO13 GND
GPIO19 / MISO CE2# / GPIO16
GPIO26 MOSI / GPIO20
GND SCLK / GPIO21
Pin 39 Pin 40
Atmega 128
12V
AC220V
5V
4채널 릴레이 모듈
5V
릴레이신호
밥통
급수/배수
UART
LED
히터봉
수위센서
VCC
GND

2 연구방법

코드설명 Raspberry Pi

```
import tkinter
import serial

ser = serial.Serial('/dev/ttyAMA0',9600) // Serial 통신
tsensor='/sys/bus/w1/devices/28-03089794462b/w1_slave' // 온도센서
```

```
global a
global b
global c
global d
a=b=c=d='0'
```

온도 설정 관련
변수를 0으로 초기화

```
def WaterINClick():
    ser.write(b'A')

def WaterOUTClick():
    ser.write(b'B')

def FeedClick():
    ser.write(b'C')

def LedONClick():
    ser.write(b'D')
```

버튼 클릭 시 작동
ATMega로 'A~E' 전송

```
def LedOFFClick():
    ser.write(b'E')
```

```
def T20():
    global a
    a = '1'
    if int(settemp())<20:
        ser.write(b'F')


def T23():
    global b
    b = '1'
    if int(settemp())<23:
        ser.write(b'F')

def T25():
    global c
    c = '1'
    if int(settemp())<25:
        ser.write(b'F')

def T27():
    global d
    d = '1'
    if int(settemp())<27:
        ser.write(b'F')
```

온도가 설정한 값보다
낮으면 변수가 1로 변경
되고 ATMega로 'F' 전송

2 연구방법

코드설명 Raspberry Pi

온도 설정 버튼
크기 / 위치 설정

```
def TEMPBOX():
    newwindow = tkinter.Toplevel(window)
    newwindow.geometry("800x480+0+0")
    t20 = tkinter.Button(newwindow, text = "20", font=('Roman, Sans',25),
command = T20)
    t23 = tkinter.Button(newwindow, text = "23", font=('Roman, Sans',25),
command = T23)
    t25 = tkinter.Button(newwindow, text = "25", font=('Roman, Sans',25),
command = T25)
    t27 = tkinter.Button(newwindow, text = "27", font=('Roman, Sans',25),
command = T27)

    t20.pack(side = "left", fill="both",expand=True)
    t23.pack(side = "left", fill="both",expand=True)
    t25.pack(side = "left", fill="both",expand=True)
    t27.pack(side = "left", fill="both",expand=True)
```

온도센서를 통해
입력받은 값으로
온도 설정해주는 함수

```
def traw():
    f=open(tsensor,'r')
    lines=f.readlines()
    f.close()
    return lines

def readtemp():
    lines=traw()
    while lines[0].strip()[-3:]!='YES':
        time.sleep(0.2)
        lines=traw()
    tout=lines[1].find('t=')
    if tout!=-1:
        tstr=lines[1].strip()[tout+2:]
        tc = round(float(tstr)/1000,1)
        space=('C')
        return tc,space

def settemp():
    lines=traw()
    while lines[0].strip()[-3:]!='YES':
        time.sleep(0.2)
        lines=traw()
    tout=lines[1].find('t=')
    if tout!=-1:
        tstr=lines[1].strip()[tout+2:]
        tc = round(float(tstr)/1000,1)        return tc
```

2 연구방법

코드설명 Raspberry Pi

```
def hitteroff():
    global a
    global b
    global c
    global d

    if int(settemp())<23:
        a = '0';

    if int(settemp())>25:
        b = '0';

    if int(settemp())>27:
        c = '0';

    if int(settemp())>30:
        d = '0';

    if a=='0' and b=='0' and c=='0' and d=='0':
        ser.write(b'G')
```

설정한 온도보다 높아지면 a b c d 값이 0 이되며 ATMega로 'G' 전송

```
def on_alarm(top_level_window):
        global var
        var.set(readtemp())
        hitteroff()

        top_level_window.after(1000,on_alarm,top_level_window)
```

현재 온도를 저장

```
window=tkinter.Tk()
window.title("Aquarium")
window.resizable(True,True)
var=tkinter.StringVar()
label=tkinter.Label(window,font=('Roman, Sans',40),fg='red',textvariable = var)
label.pack(anchor="nw",fill="x")
WaterIN = tkinter.Button(window, text = "Water IN", font=('Roman,
Sans',20),command = WaterINClick)
WaterOUT = tkinter.Button(window, text = "Water OUT", font=('Roman,
Sans',20),command = WaterOUTClick)
Feed = tkinter.Button(window, text = "FEED", font=('Roman,
Sans',20),command = FeedClick)
LedON = tkinter.Button(window, text= "LedON", font=('Roman,
Sans',20),command = LedONClick)
LedOFF = tkinter.Button(window, text= "LedOFF", font=('Roman,
Sans',20),command = LedOFFClick)
HITTER = tkinter.Button(window, text= "SetTEMP", font=('Roman,
Sans',20),command = TEMPBOX)
WaterIN.pack(side="left",fill="both",expand=True)
WaterOUT.pack(side="left",fill="both",expand=True)
Feed.pack(side="left",fill="both",expand=True)
LedON.pack(side="left",fill="both",expand=True)
LedOFF.pack(side="left",fill="both",expand=True)
HITTER.pack(side="right",fill="both",expand=True)
window.after(1000, hitteroff)
window.after(1000, on_alarm, window)
window.mainloop()
```

현재 온도를 화면에 표시

화면에 원하는 동작을 하기위한 버튼을 만듬

2 연구방법

코드설명 CodeVision AVR

```
#include <mega128.h>
#include <delay.h>
#include <stdio.h>

#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

#pragma used+
char getchar1(void)
{
char status,data;
while (1)
      {
      while (((status=UCSR1A) & RX_COMPLETE)==0);
      data=UDR1;
      if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
         return data;
      }
}
#pragma used-
```

2 연구방법

코드설명 CodeVision AVR

```
#pragma used+
void putchar1(char c)
{
while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
UDR1=c;
}
#pragma used-

#define FIRST_ADC_INPUT 0
#define LAST_ADC_INPUT 1
unsigned int adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];
#define ADC_VREF_TYPE 0x00

interrupt [ADC_INT] void adc_isr(void)
{
static unsigned char input_index=0;
// Read the AD conversion result
adc_data[input_index]=ADCW;
// Select next ADC input
if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))
   input_index=0;
ADMUX=(FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff))+input_index;
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
}
```

```
void main(void)
{
int LOW;
int HIGH;
PORTA=0x00;
DDRA=0x1E;

PORTB=0x00;
DDRB=0x14;

UCSR1A=0x00;
UCSR1B=0x18;
UCSR1C=0x06;
UBRR1H=0x00;
UBRR1L=0x67;


UCSR0A=0x00;
UCSR0B=0x18;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x67;

ADMUX=FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff);
ADCSRA=0xCC;

#asm("sei")
```

ADC 0,1
PORTA.1~4 = 릴레이 사용
PORTB.3,5 = 배수펌프
UART0 = CodeVision Terminal
UART1 = Raspberry Pi

2 연구방법

코드설명 CodeVision AVR

```
PORTA=0x1E;

while (1)
    {
    LOW = adc_data[0];
    HIGH = adc_data[1];


    if(LOW < 50)
    {
    PORTB=0x00;
    }
    if(HIGH>950)
    {
    PORTA.4=0x01;
    }
```

```
switch(getchar1())
    {
    case 'A':
    printf("Press A₩n₩r");
    if(HIGH<900)
    PORTA.4=0x00;
    printf("START Water IN!!₩n₩r");
    break;

    case'B':
    printf("Press B₩n₩r");
    if(LOW>900)
    {
    PORTB=0x14;

    printf("START Water OUT!!₩n₩r");
    }
    else
    printf("Little Water₩n₩r");

    break;
```

```
    case 'C':
    printf("Press C₩n₩r");
    PORTA.3=0x00;
    delay_ms(500);
    PORTA.3=0x01;
    printf("FEED!!₩n₩r");
    break;

    case'D':
    printf("Press D₩n₩r");
    PORTA.1=0x00;
    printf("LED ON!!₩n₩r");

    break;


    case 'E':
    printf("Press E₩n₩r");
    PORTA.1=0x01;
    printf("LED OFF!!₩n₩r");

    break;
```

```
    case 'F':
    printf("Press E₩n₩r");
    PORTA.2=0x00;
    printf("HITTER ON !!₩n₩r");

    break;

    case 'G':
    printf("Press G₩n₩r");
    PORTA.2=0x01;
    printf("HITTER OFF!!₩n₩r");

    break;

    }

    delay_ms(2000);
    }
}
```

3 연구결과

외관사진

3 연구결과

동작사진

[급수펌프 동작 전/후]

3 연구결과

동작사진

[배수펌프 동작 전/후]

3 연구결과

동작사진

[LED 동작 전/후]

3 연구결과

동작사진

[사료 공급 장치 동작 사진]

3 연구결과

동작사진

[온도센서 히터봉 동작으로 기포가 올라오면서 온도 상승]

감사합니다