

Instrumentation

သတိ : စာကိုမကျက်ပါနှင့်။
နားလည်အောင်ဖတ်ပြီးစဉ်းစားပါ။

Smart Devices and Embedded Systems

- အခုခေတ်ကြီးမှာ Google Home Assistant လို Smart Devices များက အင်မတန် ခေတ်စားလာပါသည်။
- ဒီလို Smart Device များသည် IoT (Internet of Things) ဆိုပြီး ခေတ်တစ်ခေတ်ကို ဆန်းသစ်စေပါသည်။
- အမှန်တော့ Smart Device များကို အရင်တုန်းက Embedded Systems များဟု လူသိများကြပါသည်။ သို့သော် လူတွေနဲ့ တိုက်ရိုက် ထိတွေ့မှုတော့ နည်းပါးခဲ့သည်။
- Embedded Systems များ၏ အသက်သည် Transducers နှင့် Microcontrollers များ ဖြစ်ပါသည်။
- သာမန် Embedded Systems များနှင့် IoT Device များ ကွာခြားသည့် အချက်ကတော့ IoT Device များက Network (Internet) ဖြင့် ချိတ်ဆက်ဆောင်ရွက်နိုင်ပါသည်။



Google Home Assistant

Embedded Systems



- Transducers များကို
 - Sensors (ဥပမာ၊ Microphone)
 - Actuators (ဥပမာ၊ Servo Motor)
- Microcontroller များက Processor အသေးစားများ ဖြစ်ပါသည်။ ဥပမာ၊ Intel™ Atom, Raspberry Pi™, Arduino™ ။

Transducers

Sensors	
Input Signal	Output Signal
Mechanical	Analog
Optical	
Thermal	Digital
Acoustical	
Chemical	

Actuators	
Input Signal	Output Signal
Analog	Mechanical
	Optical
Digital	Thermal
	Acoustical
	Chemical

- Transducers များကို
 - Sensors (ဥပမာ၊ Microphone)
 - Actuators (ဥပမာ၊ Servo Motor)
- Sensors များသည် Environment မှ Signal များကို Measure လုပ်ပြီး Electrical Signal (Analog or Digital) အဖြစ်သို့ ပြောင်းလဲပေးပါသည်။
- Actuators များက Electrical Signal (Analog or Digital) များကို Environment နှင့် Interface (Interact) လုပ်ဖို့ သက်ဆိုင်ရာ Signal များ အဖြစ်သို့ ပြောင်းလဲပေးပါသည်။

Microcontroller

- Sensors များမှ ရရှိလာသော Electrical Signal များကို လိုအပ်သလို Process လုပ်ပါသည်။
- Processed လုပ်ပြီးသော Output Electrical Signal များကို သက်ဆိုင်ရာ Actuators များသို့ ပို့ပါသည်။
- IoT Device များတွင် Sensors များ၊ Microcontroller များ နှင့် Actuators များသည် Device တစ်ခုထဲမှာ ရှိချင်မှ ရှိမည် ဖြစ်ပြီး ထို Devices များ အားလုံးကို Network သို့မဟုတ် Internet ဖြင့် ချိတ်ဆက်ထားနိုင်ပါသည်။

Errors

- Sensors များသည် Environment မှ Signal များကို Capture လုပ်ပြီး Measurement များ လုပ်ဆောင်ရာတွင် အမြဲတမ်း Accuracy နှင့် Precision မရှိနိုင်ဘဲ Error များ ရှိနိုင်ပါသည်။ ဥပမာ၊ Speech Recognition အတွက် Microphone ဖြင့် အသံဖမ်းရာတွင် အမြဲ ကြည်လင်ပြတ်သား မည်မဟုတ်ပါ။
- Error များကို Systematic Error နှင့် Random Error များ ဟူ၍ ခွဲခြားနိုင်ပါသည်။
 - Systematic Error များသည် အများအားဖြင့် Sensor ကြောင့် သို့မဟုတ် Environment ကြောင့် ဖြစ်သော Error များ ဖြစ်ပါသည်။
 - Random Error များကတော့ အများအားဖြင့် Unpredictable Events (Noises) များကြောင့် ဖြစ်သော Error များ ဖြစ်ပါသည်။
- Systematic Error များကို အများအားဖြင့် Calibration (Tuning) လုပ်ခြင်းဖြင့် ဖြေရှင်းကြပြီး Random Error များကို တော့ Statistical Method များဖြင့် ဖြေရှင်းကြပါသည်။

SNR (Signal to Noise Ratio)

- Sensor မှ ရရှိလာသော Measurement သည် အမြဲတမ်း Electrical Signal ဖြစ်ပြီး Electrical Signal များကို Analog Signal နှင့် Digital Signal များဟူ၍ ခွဲခြားနိုင်ပါသည်။
- Electrical Signal များတွင် Data Signal (V_s) များသာမက Noise Signal (V_n) များလည်း ပါဝင်နေနိုင်ပါသည်။
- Signal Quality ကို ကျွန်တော်တို့ SNR ဖြင့် ဖော်ပြလေ့ရှိပါသည်။

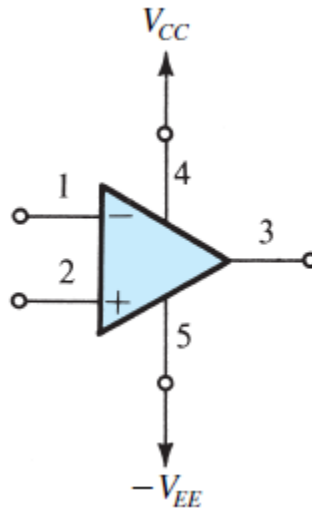
$$SNR = 20\log_{10} \left(\frac{V_s}{V_n} \right)$$

- Noise Signal များကို ကျွန်တော်တို့ Earthing လုပ်ခြင်း၊ Shielding လုပ်ခြင်းတို့ဖြင့် Physically Reduce လုပ်လို့ ရပါသည်။ ဒါ့အပြင် Signal Processing ကိုသုံးပြီးတော့လည်း Reduce လုပ်နိုင်ပါသည်။

Operational Amplifier (Op Amp)

- Sensor မှ ရရှိလာသော အချို့ Signal များသည် တဖြည်းဖြည်း Attenuation ဖြစ်သွားတတ်ပါသည်။ ထိုအခါ Signal များကို Amplify ပြန်လုပ်ပေးရလေ့ ရှိပါသည်။
- Operational Amplifier များသည် Signal Amplification သာမက အလွန်အသုံးဝင်သော Circuit များ ဖြစ်ပါသည်။

$$V_3 = A(V_2 - V_1)$$



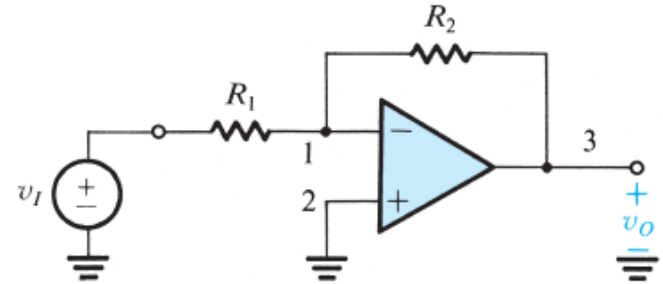
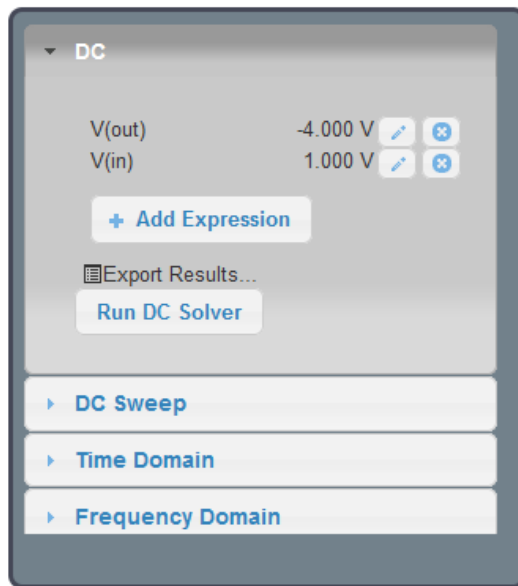
Circuit Symbol



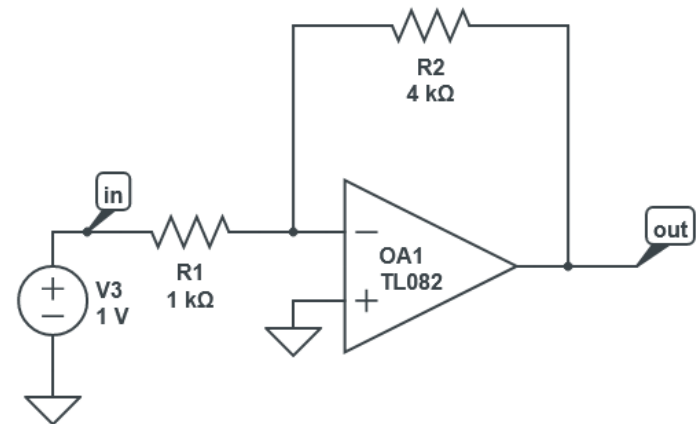
Texas Instrument Op Amp

Simple Amplifier

- ကျွန်တော်တို့ Simple Power Amplification လုပ်ချင်ရင် ဒီလို Circuit နဲ့လုပ်လို့ ရပါတယ်။
- ကျွန်တော်တို့ Circuit Lab (<https://www.circuitlab.com/>) မှာ Simulate လုပ်ကြည့်ရင် ဒီလို ရပါမည်။

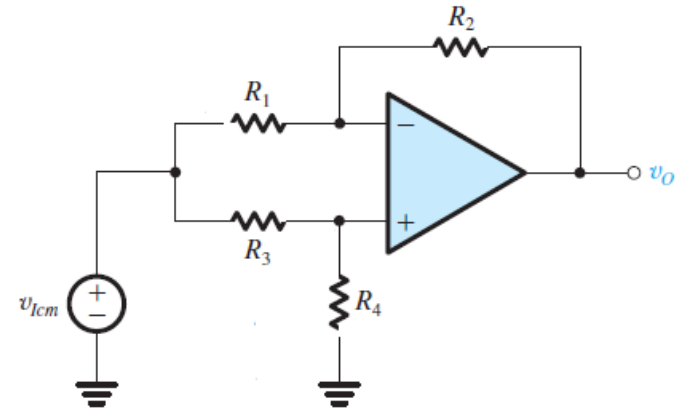


$$V_O = -\frac{R_2 V_I}{R_1}$$

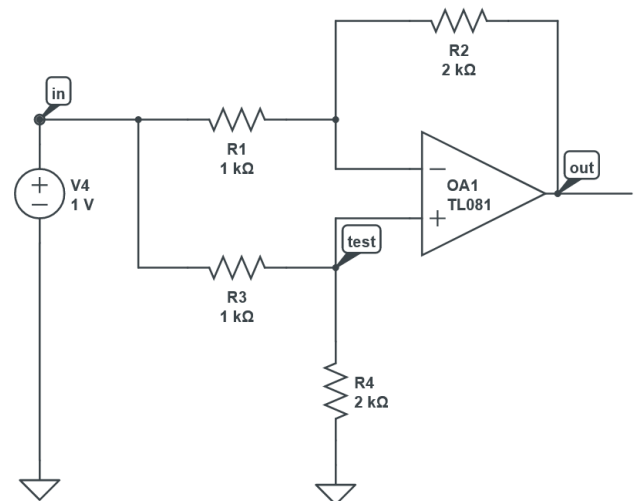
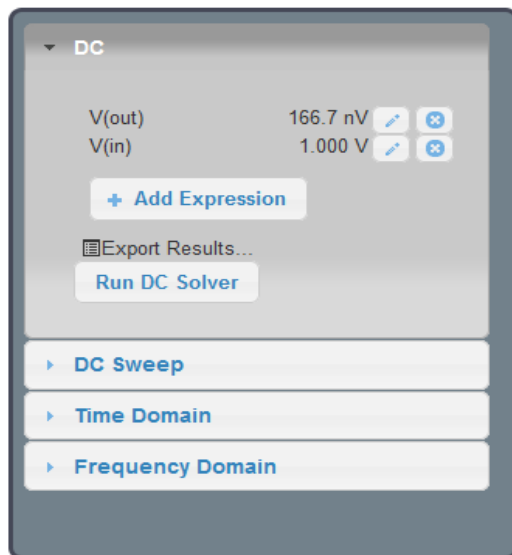


Difference Amplifier

- Difference Amplifier က Common Mode ကို Reject လုပ်ပါသည်။
- ဒီလို ဖြစ်အောင် $\frac{R_4}{R_3} = \frac{R_2}{R_1}$ ထားရပါမည်။ ဒါဆိုရင် V_{θ} သည် 0။

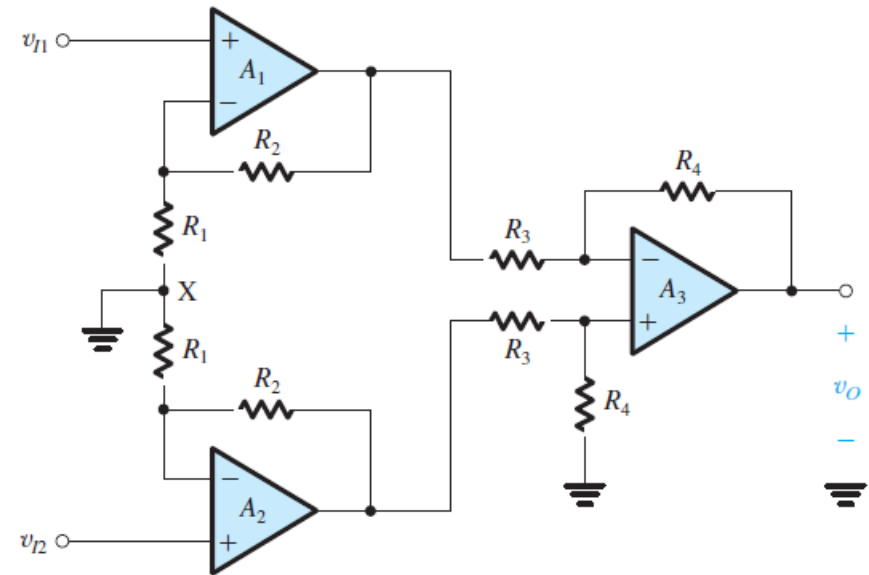


$$V_O = \left(\frac{R_4}{R_4 + R_3} \right) \left(1 - \frac{R_2 R_3}{R_1 R_4} \right) V_I$$



Instrumentation Amplifier

- Very Low Power Signal များကို Amplify လုပ်ချင်တယ်ဆိုရင်တော့ Instrumentation Amplifier ကို သုံးသင့်ပါသည်။
- Instrumentation Amplifier က Common Mode Rejection ဖြင့် Noise များကို Cancel Out ဖြစ်သွားစေပါသည်။



Analog Signal Processing

- Transducers များမှ ရလာသော Signal များသည် Electrical Signal များ ဖြစ်ပြီး Analog Signal သို့မဟုတ် Digital Signal များ ဖြစ်နိုင်ပါသည်။
- Microcontroller များသည် Digital Signal များ ဖြစ်သည့် အတွက် Transducers များမှ ရလာသော Signal များသည် Analog Signal များ ဖြစ်ပါက Digital Signal အဖြစ်သို့ ပြောင်းရလေ့ ရှိပါသည်။
- သို့သော် Analog Signal များကို Digital Signal အဖြစ်သို့ မပြောင်းခင် တခါတလေ Preprocessing လုပ်ရလေ့ ရှိပါသည်။
- ဒါကြောင့် Analog Signal Processing များကို ဆွေးနွေးပါမည်။
- Analog Signal များကို က Digital Signal အဖြစ်သို့ ပြောင်းပြီးပါက Digital Signal Processing ကို လုပ်ဆောင်ရမည် ဖြစ်သည်။ Digital Signal Processing ကိုတော့ အများအားဖြင့် Programming Code များဖြင့် Process လုပ်ကြပါသည်။
- သို့သော် Analog Signal Processing များကိုတော့ Analog Circuit များဖြင့်သာ လုပ်ဆောင်နိုင်ပါသည်။

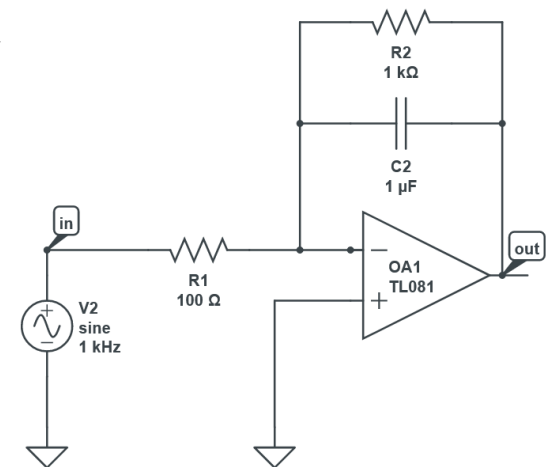
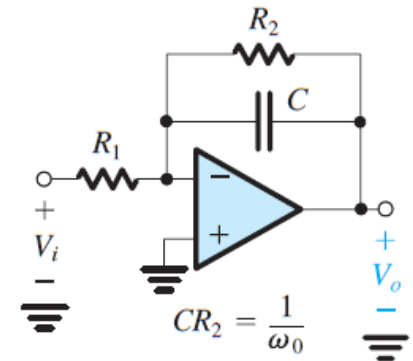
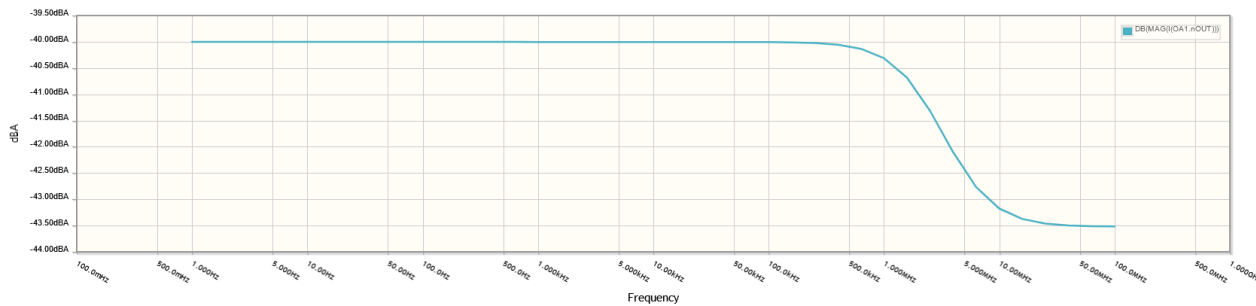
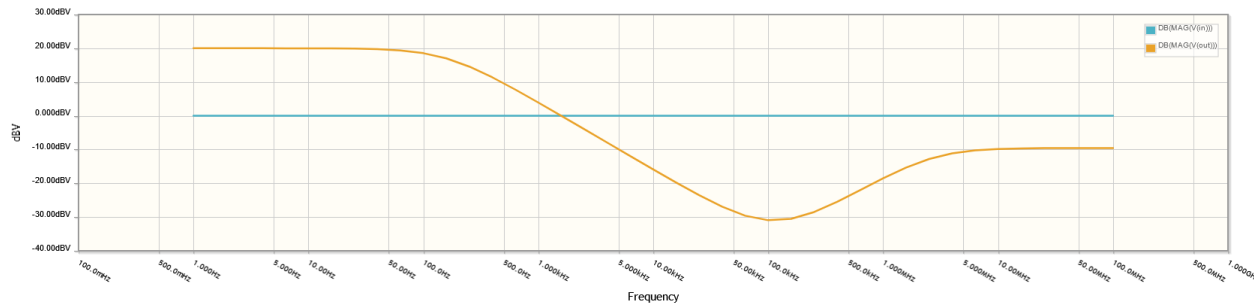
Analog Filters

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau$$

- ကျွန်တော်တို့ Signal Processing မှာ ပြောခဲ့သလို Input Signal များ၏ Frequency Component များကို Filter လုပ်လို့ ရပါသည်။ Filter များသည် Input Signal များ၏ Impulse Response များ ဖြစ်ပါသည်။ ထို့ကြောင့် Filter လုပ်ခြင်းသည် Input Signal များ၏ Convolution များ ဖြစ်ကြပါသည်။
- Filters (Analog ဝဲဖြစ်ဖြစ် Digital ဝဲဖြစ်ဖြစ်) များကို 4 မျိုးခွဲနိုင်ပါသည်။
 - Low Pass Filter (Reject High Frequency Components)
 - High Pass Filter (Reject Low Frequency Components)
 - Band Pass Filter (Pass Mid Frequency Components)
 - Band Stop Filter (Reject Mid Frequency Components)

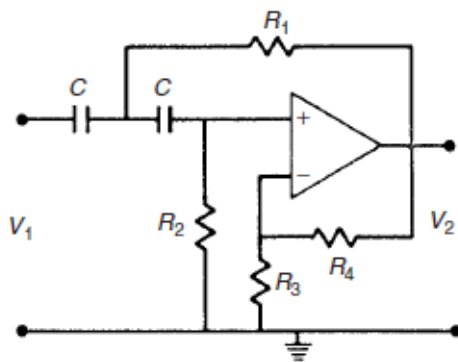
Low Pass Analog Filter

- ကျွန်တော်တို့ Magnitude Bode Plot မှာ ကြည့်ရင် High Frequency Components များကို Reject လုပ်သည်ကို တွေ့ရပါမည်။

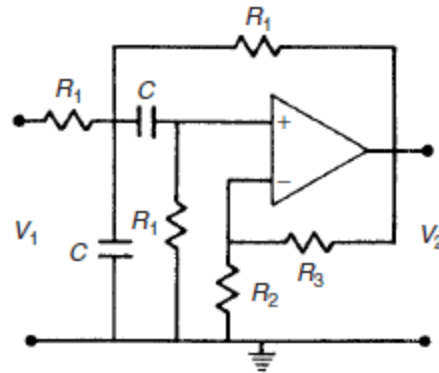


Other Analog Filters

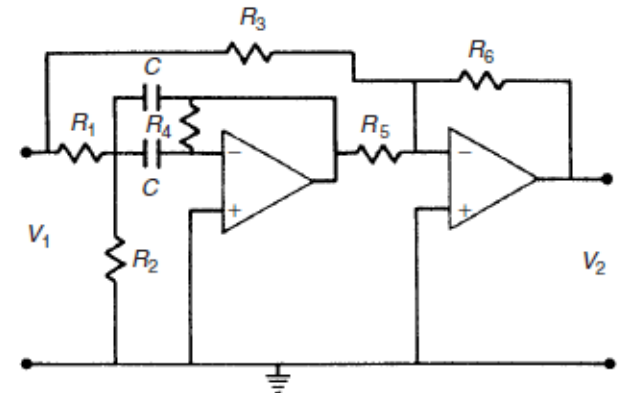
- အခြား Analog Filters များကိုလည်း Circuit Lab မှ စမ်းကြည့်ပါ။ ဒီမှာတော့ အသေးစိတ်မပြောတော့ပါဘူး။



High Pass Filter



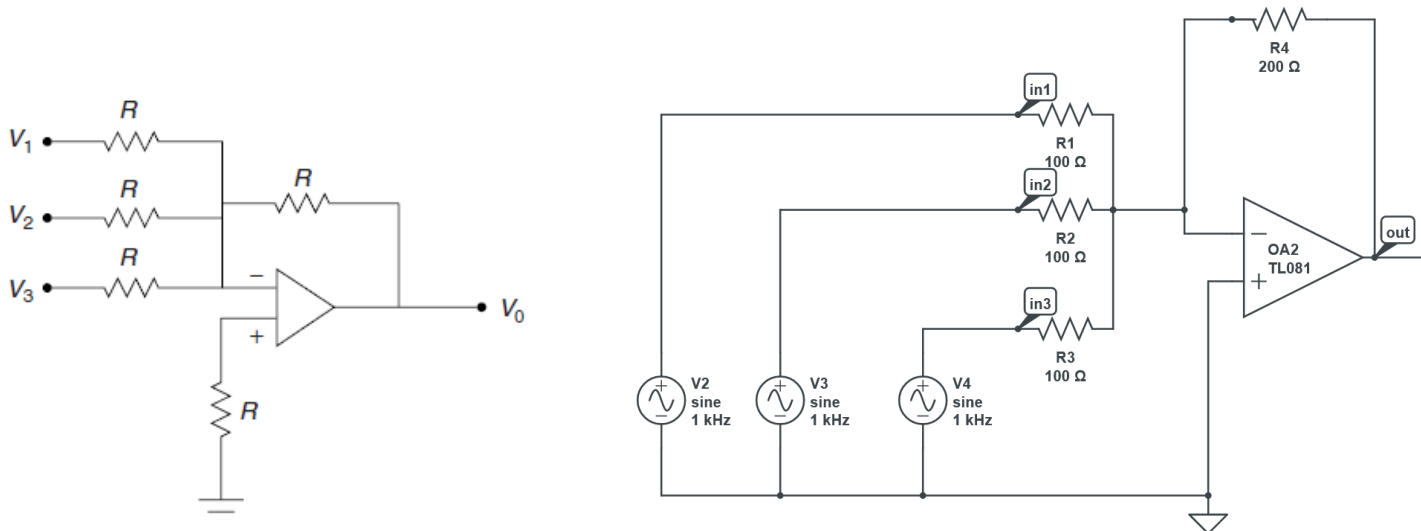
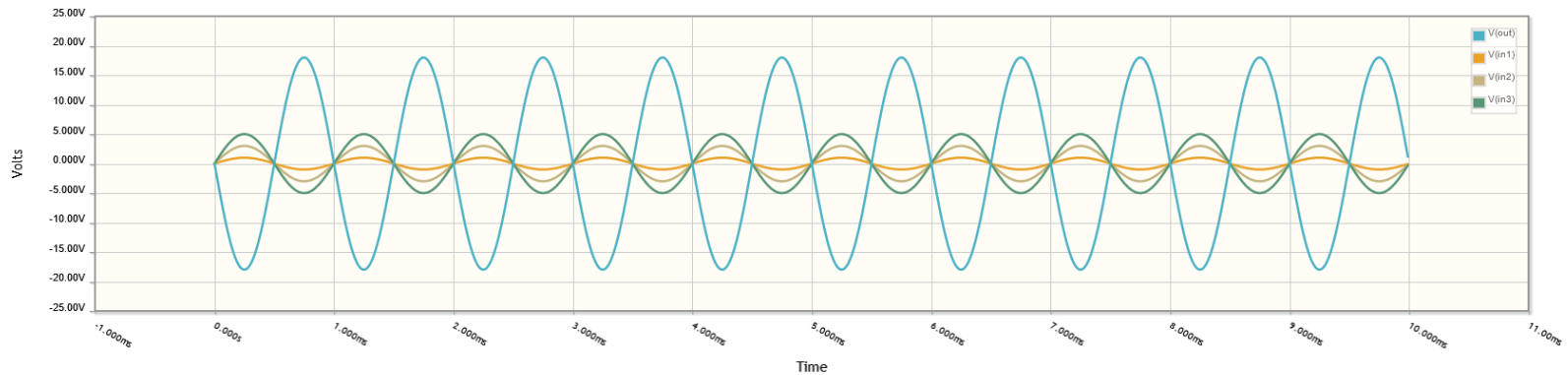
Band Pass Filter



Band Stop Filter

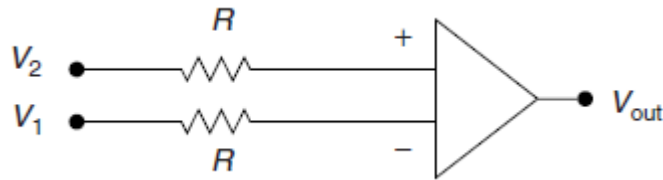
Analog Signal Addition

- Analog Signal များကို ပေါင်းချင်းလျှင် ကျွန်တော် Addition Circuit ဖြင့် လုပ်ဆောင်နိုင်ပါသည်။

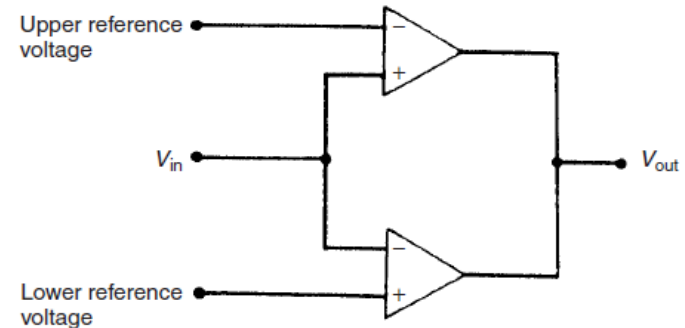


Analog Signal Subtraction

- Analog Signal များကို Subtract လုပ်ချင်လျှင် ကျွန်တော် Regular Op Amp Circuit ဖြင့် လုပ်ဆောင်နိုင်ပါသည်။ Common Mode Rejection အရ V_{out} သည် အမြဲ $A(V_2 - V_1)$ ဖြစ်ပြီး ပုံမှန်အားဖြင့် $A = 10$ ဖြစ်ပါသည်။



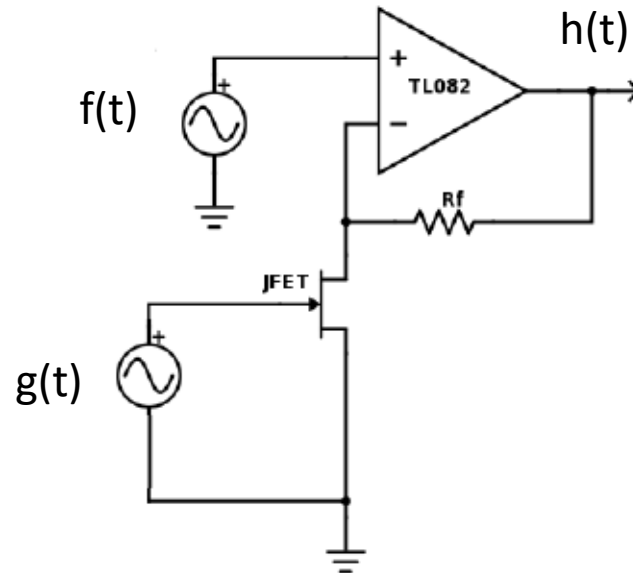
$$V_{out} = A(V_2 - V_1)$$



Analog Signal Multiplication (Modulation)

- Signal နှစ်ခု မြှောက်ခြင်းကို Amplitude Modulation ဟုခေါ်ပါသည်။ ဘာလို့လဲ ဆိုတော့ Signal နှစ်ခုကို Multiply လုပ်လျှင် Signal တစ်ခုသည် အခြား Signal ၏ Amplitude ဖြစ်သွား၍ ဖြစ်သည်။

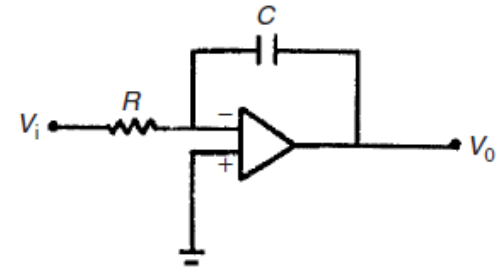
$$h(t) = g(t) \times f(t) \text{ where } g(t) \text{ is Amplitude of } f(t)$$



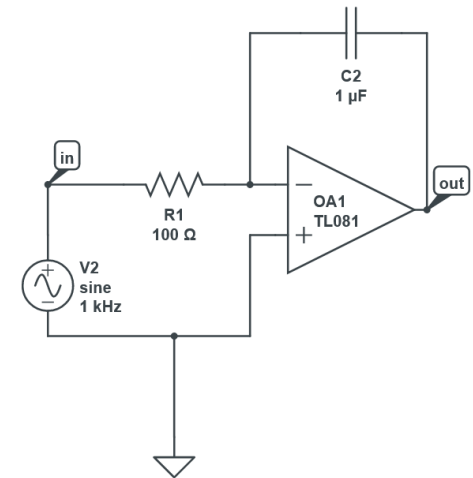
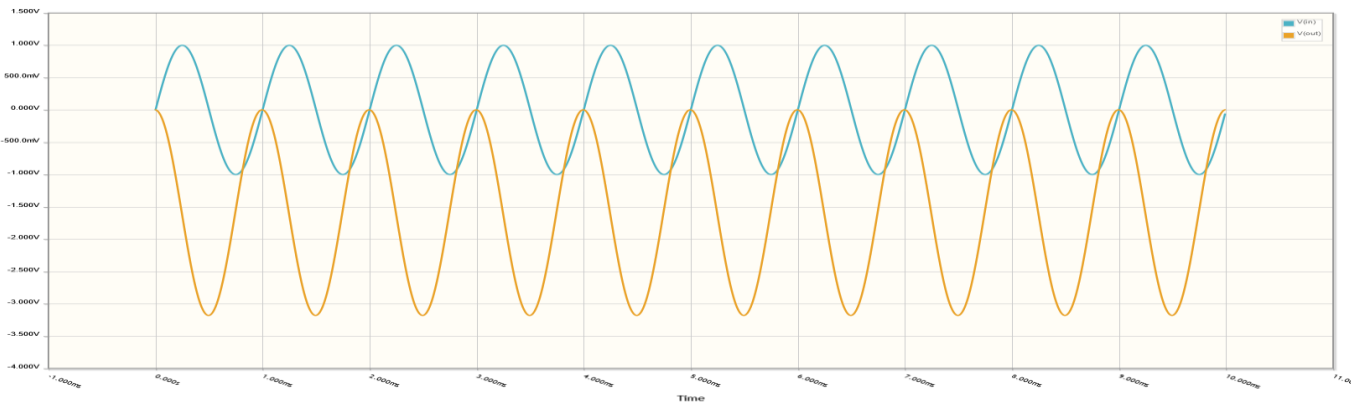
Analog Signal Integration (Accumulation)

- Analog Signal များကို Integrate (Accumulate) လုပ်ချင်ရင် Integrator Circuit ကိုသုံးပါသည်။

$$V_0 = -\frac{1}{RC} \int V_i dt$$

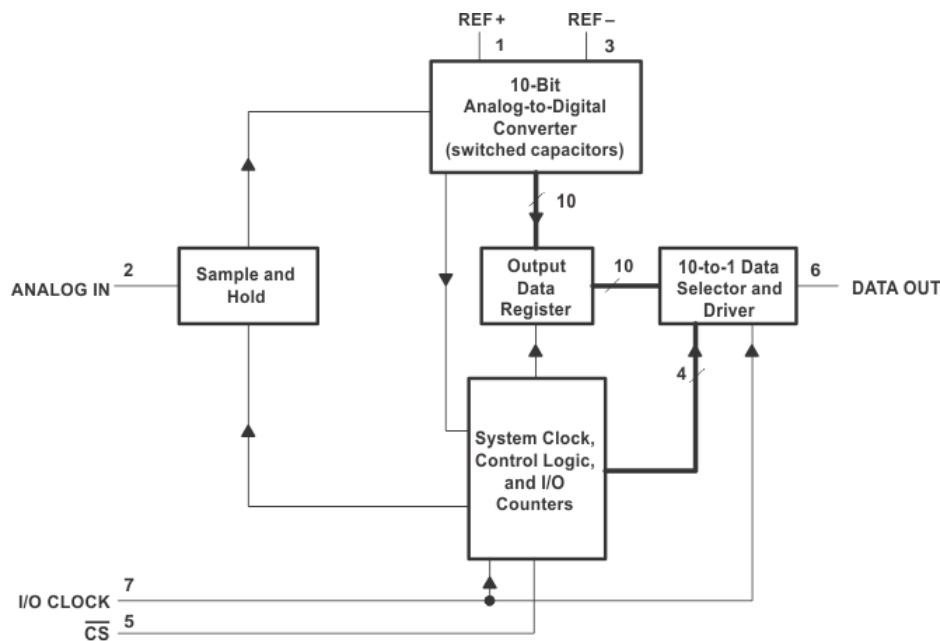


ကျွန်တော်တို့ Sine Function ကို Integrate လုပ်ရင် Cosine Function ကိုရမည် ဖြစ်သည်။



Analog to Digital Signal

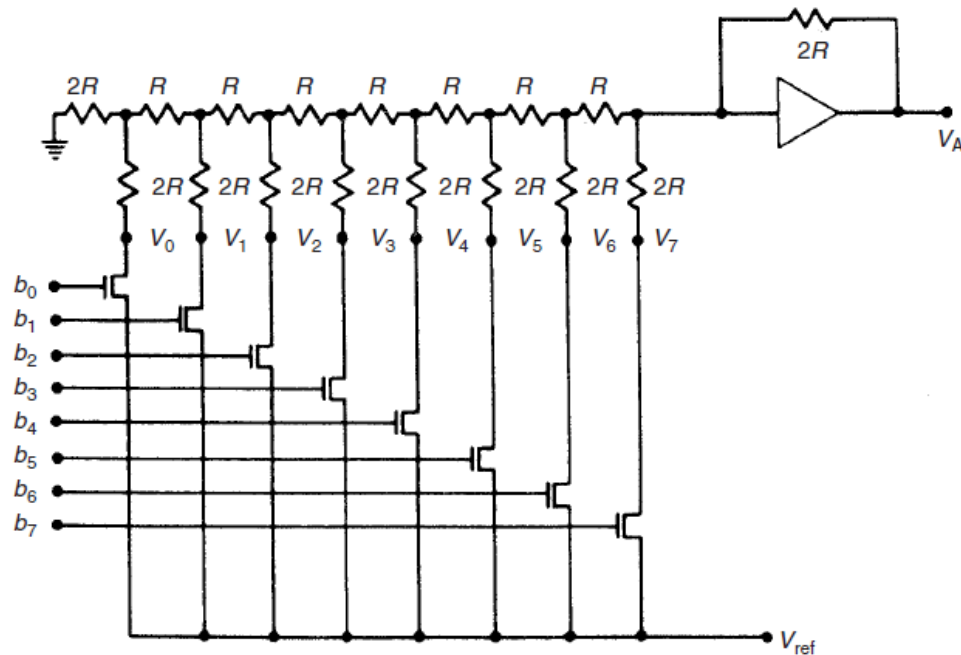
- Analog Signal များကို Digital Signal အဖြစ် ပြောင်းလိုပါက ကျွန်တော်တို့ PCM (Pulse Code Modulation) ကို သုံးကြောင်း Signal Processing မှာ ပြောခဲ့ပါသည်။
- PCM မှာ 2 ဝိုင်းပါပါသည်။ ပထမက Sampling ဖြစ်ပြီး ဒုတိယက Quantization ဖြစ်ပါသည်။ A/D Converter များမှာ အကြမ်းအားဖြင့်



DAC7811

Digital to Analog Signal

- Digital Signal များကို Analog Signal အဖြစ် ပြောင်းလိုပါက ကျွန်တော်တို့ အောက်ပါ Circuit များကို တည်ဆောက်နိုင်ပါသည်။

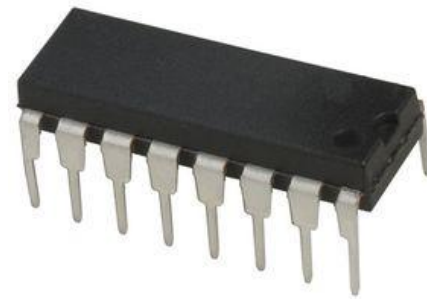
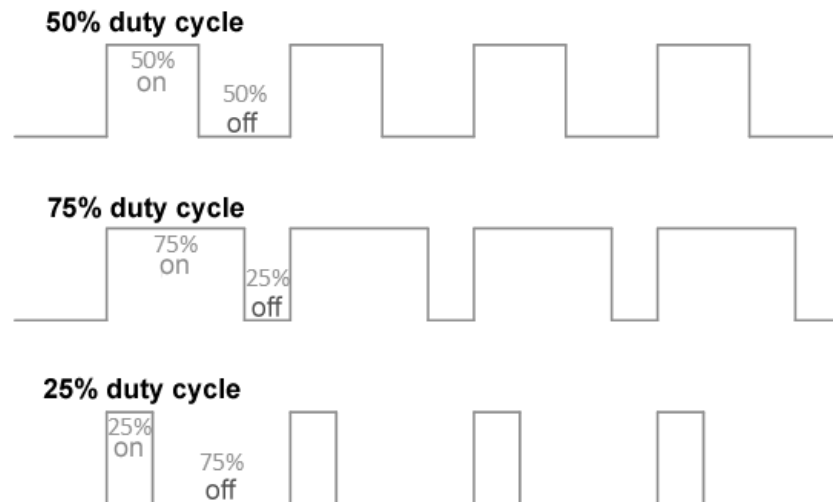


Digital Signal Processing

- အများအားဖြင့် Digital Signal များကိုတော့ Microcontroller သို့မဟုတ် Microprocessor ပေါ်မှာ Program Code များဖြင့် Process လုပ်ကြပါသည်။
- Microcontroller သို့မဟုတ် Microprocessor ပေါ်မူတည်ပြီး အသုံးပြုရမည့် Programming Language များတော့ ကွာခြားမည် ဖြစ်သည်။
- Raspberry Pi™ ပေါ်တွင် အများအားဖြင့် Python ကိုသုံးအပြုပါသည်။ GPIO Library သည် အသုံးများသော Library ဖြစ်ပါသည်။
- <https://gpiozero.readthedocs.io/en/stable/installing.html>
- Microcontroller မှ Output Digital Signal များကိုတော့ သက်ဆိုင်ရာ Actuator များသို့ တိုက်ရိုက်ဖြစ်စေ၊ Analog Signal သို့ ပြောင်းပြီး ဖြစ်စေ လုပ်ဆောင်ပါသည်။

Pulse Width Modulation

- ကျွန်တော်တို့ Drone တစ်ခုအတွက် Servo Motor ကို Microcontroller မှ Control လုပ်မည်ဆိုပါက Servo Motor ရဲ့ RPM (Revolution Per Minute) သည် Motor ကို ပေးသော DC Voltage ပေါ်မူတည်ပါသည်။
- Voltage များများပေးရင် Motor မြန်မြန်လည်ပြီး Voltage နည်းနည်းပေးရင် Motor ဖြည်းဖြည်းလည်မည် ဖြစ်သည်။
- ဒါကို ကျွန်တော်တို့ Power Duty Cycle ဖြင့် Control လုပ်လိုရပြီး ဒီလိုလုပ်ခြင်းကို Pulse Width Modulation ဟုခေါ်ပါသည်။



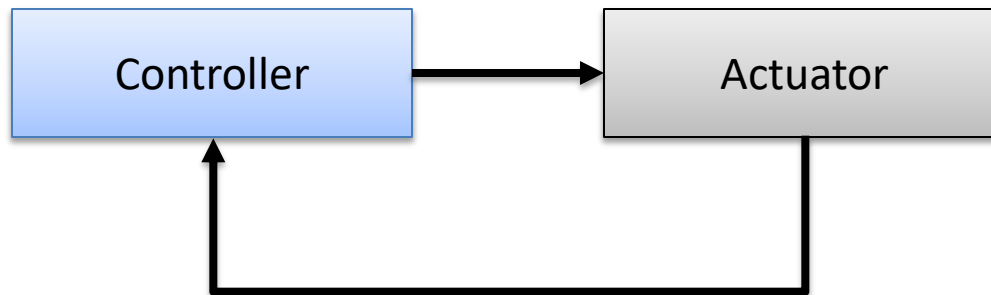
Pulse Code Modulator KA3525A

Open Loop and Close Loop Control

- Actuator များကို Control လုပ်ရာတွင် Open Loop Control နှင့် Close Loop Control ဆိုပြီး 2 မျိုး ရှိပါသည်။
- Open Loop Control တွင် Feedback Loop မပါသည့် အတွက် Error Correction ကို မလုပ်နိုင်ပါ။
- Close Loop Control တွင် Feedback Loop ပါသည့် အတွက် Error Correction ကို လုပ်နိုင်ပါသည်။



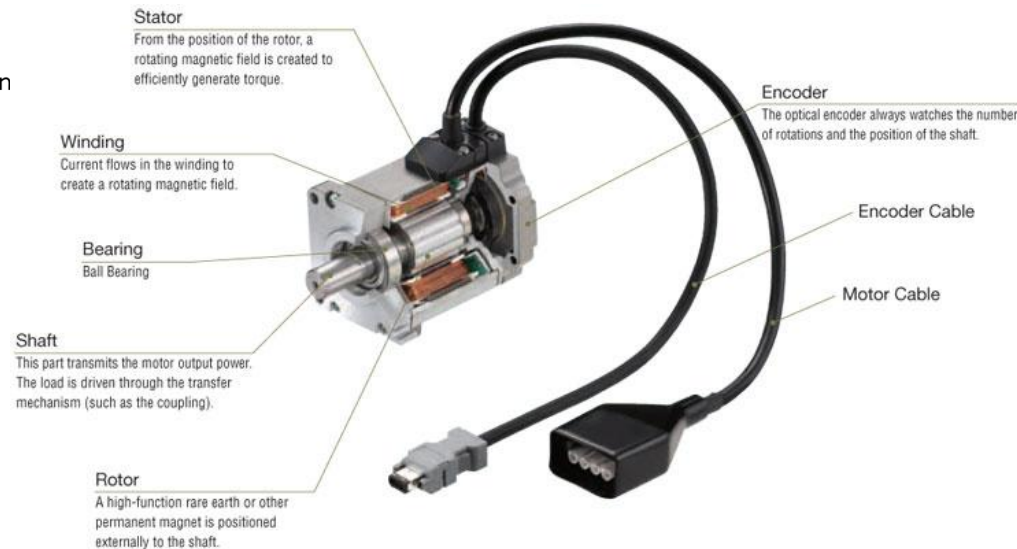
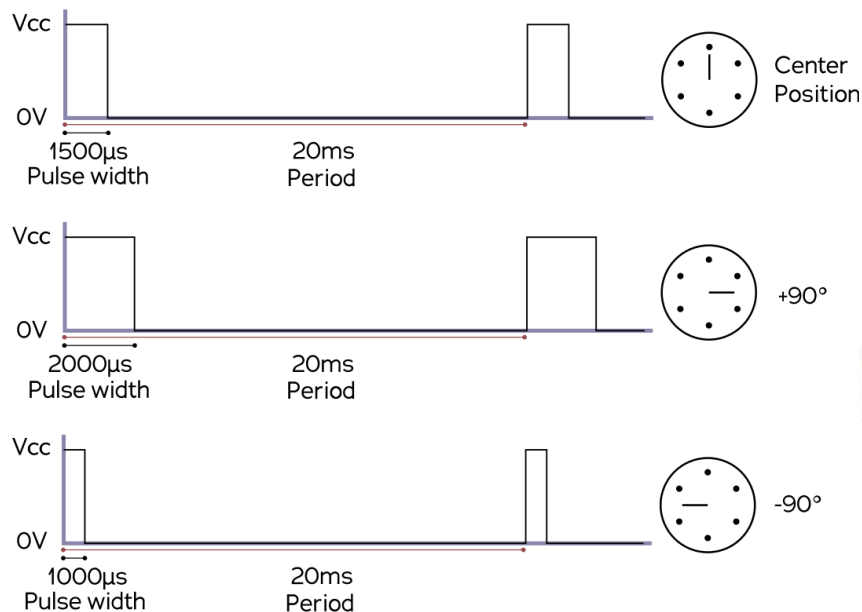
Open Loop Control



Close Loop Control

Servo Control with PWM

- အများအားဖြင့် Servo Motor များသည် Close Loop Control (with Feedback Loop) ဖြစ်ပြီး Motor Cable မှ PWM များဖြင့် Motor ၏ RPM ကို Control လုပ်ပါသည်။
- Optical Shaft Encoder မှ Motor ၏ Rotation နှင့် Position ကို Feedback ပြန်ပေးပါသည်။



Summary

- အခုဆိုရင်တော့ Instrumentation အကြောင်းနည်းနည်း နားလည်သွားလောက်ပြီလို့ ထင်ပါသည်။
- အသေးစိတ်ကိုတော့ Instrumentation စာအုပ် ဖတ်ကြည့်သင့်ပါသည်။ ပြီးလျှင် Circuit များကို Circuit Lab မှာ Simulator ဖြင့် စမ်းကြည့်ပါ။
- ဖြစ်နိုင်လျှင်တော့ Physical Components များဖြင့် လက်တွေ့ စမ်းနိုင်လျှင် ပိုကောင်းပါသည်။
- အမှန်တော့ Instrumentation သည် Embedded System များ၊ IoT Device များ၊ Robot များ နှင့် Automation System များ လုပ်ရာတွင် မရှိမဖြစ် လိုအပ်ပါသည်။

Appendix-A Rice Cooker

- ထမင်းချက်ခြင်းသည် Computational Problem (တွက်ချက်ခြင်း) ဟု ကျွန်တော် အရင်က ပြောခဲ့ဖူးပါသည်။ အမှန်တော့ လူတိုင်း Rice Cooker ဖြင့် ထမင်းချက် ဖူးမည် ထင်ပါသည်။ သို့သော် Rice Cooker ဘယ်လို အလုပ်လုပ်လဲတော့ စိတ်ဝင်စားတဲ့သူ နည်းပါမည်။
- အမှန်တော့ Rice Cooker ကို ဂျပန်တွေ စတင်ခဲ့တာ ဖြစ်ပြီး အင်မတန် အသုံးများလာခဲ့ပါသည်။
- သဘောတရားကတော့ ရိုးရိုးလေး ဖြစ်သည်။ ဆန်ကို ရေနဲ့ ပြုတ်ပါသည်။ ထိုအခါ Thermostat က အပူချိန်ကို တိုင်းပါသည်။ ထမင်းအိုးထဲမှာ ရေမခမ်းသ၍ အပူချိန်သည် 100 Degree Celsius အောက်မှာ ရှိပါသည်။ ထိုအခါ အပူ ဆက်ပြီးပေးပါမည်။ သို့သော် ရေခမ်းသည်နှင့် အပူချိန် တဖြည်းဖြည်းတက်လာပြီး 100 Degree Celsius ကို ကျော်လာပါမည်။ ထိုအခါ အပူပေးခြင်းကို ရပ်ပါမည်။ ထမင်းအိုး ကျက်ပါပြီ။ အင်မတန် ရိုးရှင်းပြီး လက်တွေ့ အသုံးဝင်သော Invention တစ်ခု ဖြစ်ပါသည်။
- ဒါကို ဘယ်လို Circuit များ တည်ဆောက်ရင် ရမလဲ။ စဉ်းစားကြည့်ပါ။



Sony's First Rice Cooker

```
while (cooking)
{
    if (temperature <= 100)
    {
        switch = true;
        cooking = true;
    }
    else
    {
        switch = false;
        cooking = false;
    }
}
```