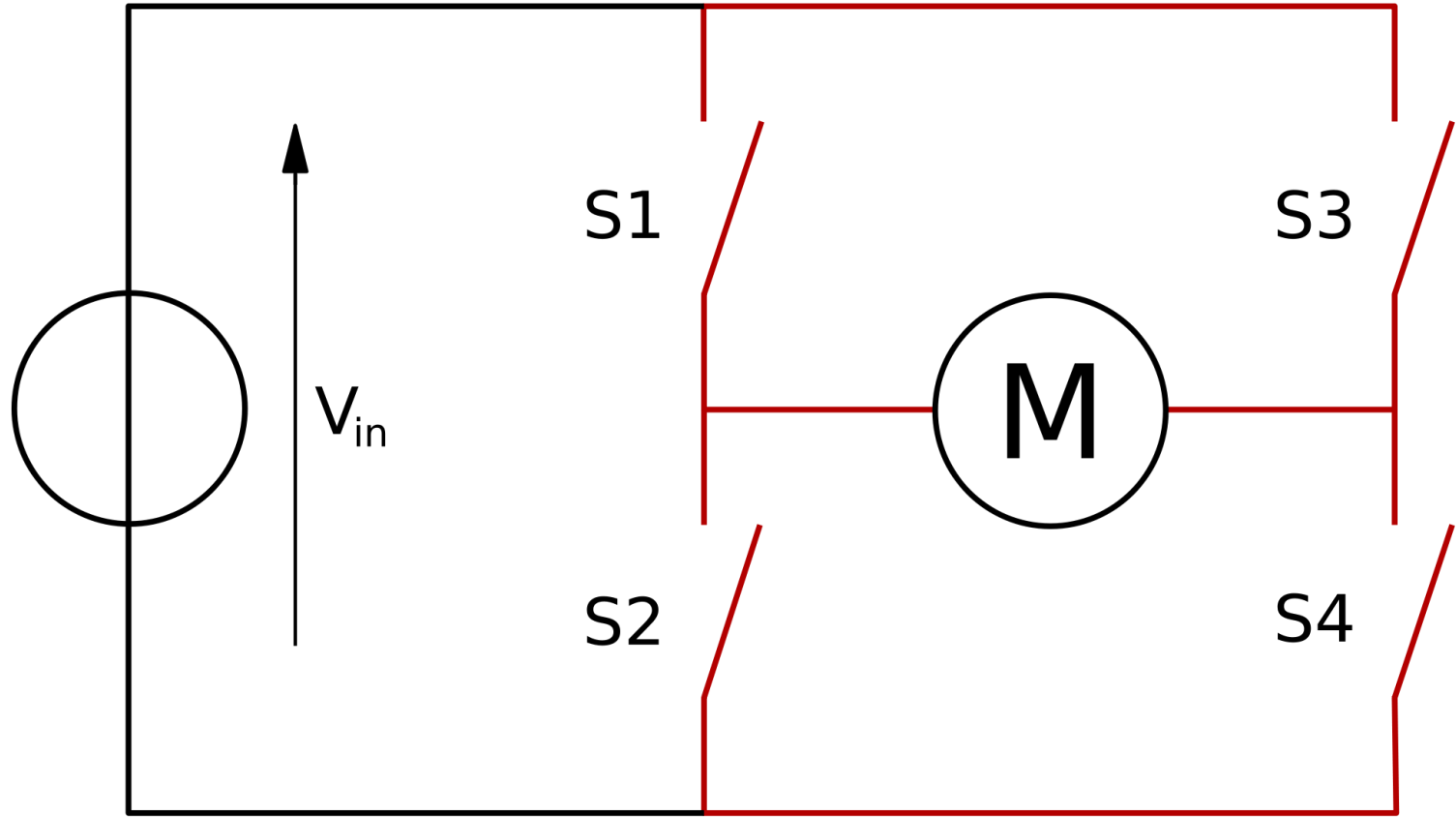
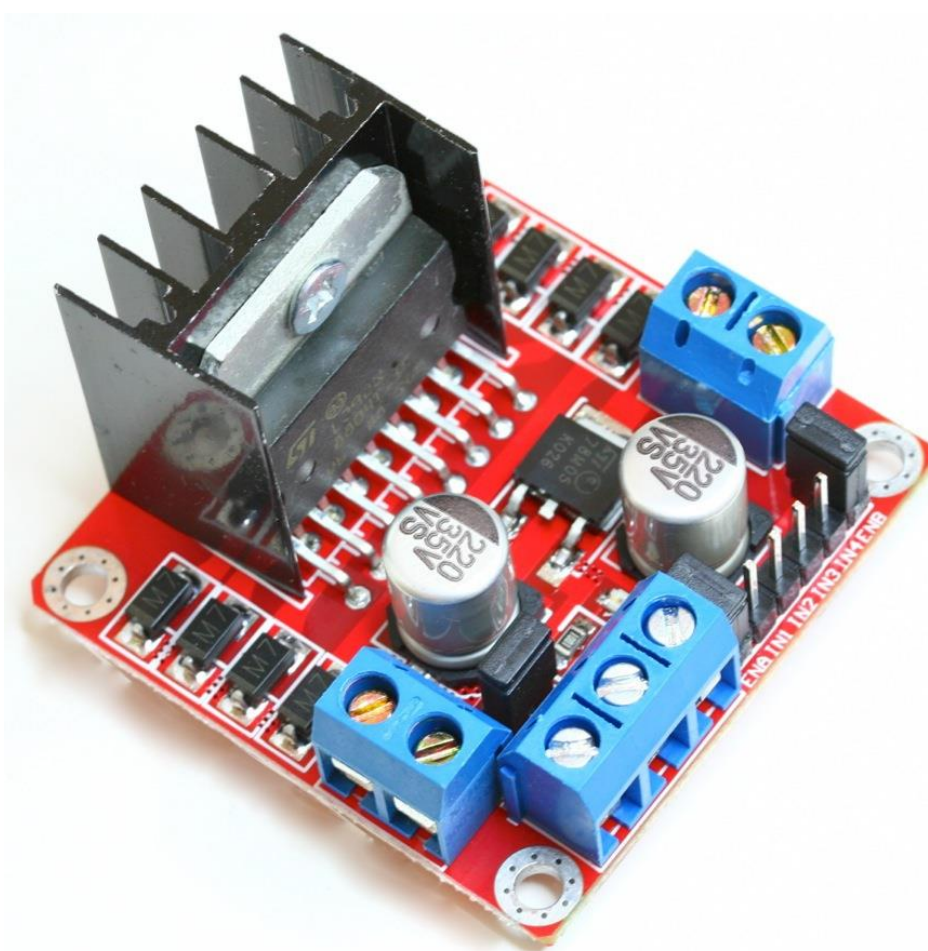


H híd



L298N Dual H Bridge



```
int A_MotorEnable = 2;
int A_MotorDir1 = 3;
int A_MotorDir2 = 4;
int A_RPM=0;

void setup() {
  pinMode(A_MotorDir1, OUTPUT);
  pinMode(A_MotorDir2, OUTPUT);
  pinMode(A_MotorEnable, OUTPUT);
}

void loop() {
  A_motorF();
  while(A_RPM<255){
    analogWrite(A_MotorEnable,A_RPM);
    delay(50);
    A_RPM++;
  }
  while(A_RPM>0){
    analogWrite(A_MotorEnable,A_RPM);
    delay(50);
    A_RPM--;
  }
}
```

```
A_motorR();
while(A_RPM<255){
  analogWrite(A_MotorEnable,A_RPM);
  delay(50);
  A_RPM++;
}
while(A_RPM>0){
  analogWrite(A_MotorEnable,A_RPM);
  delay(50);
  A_RPM--;
}

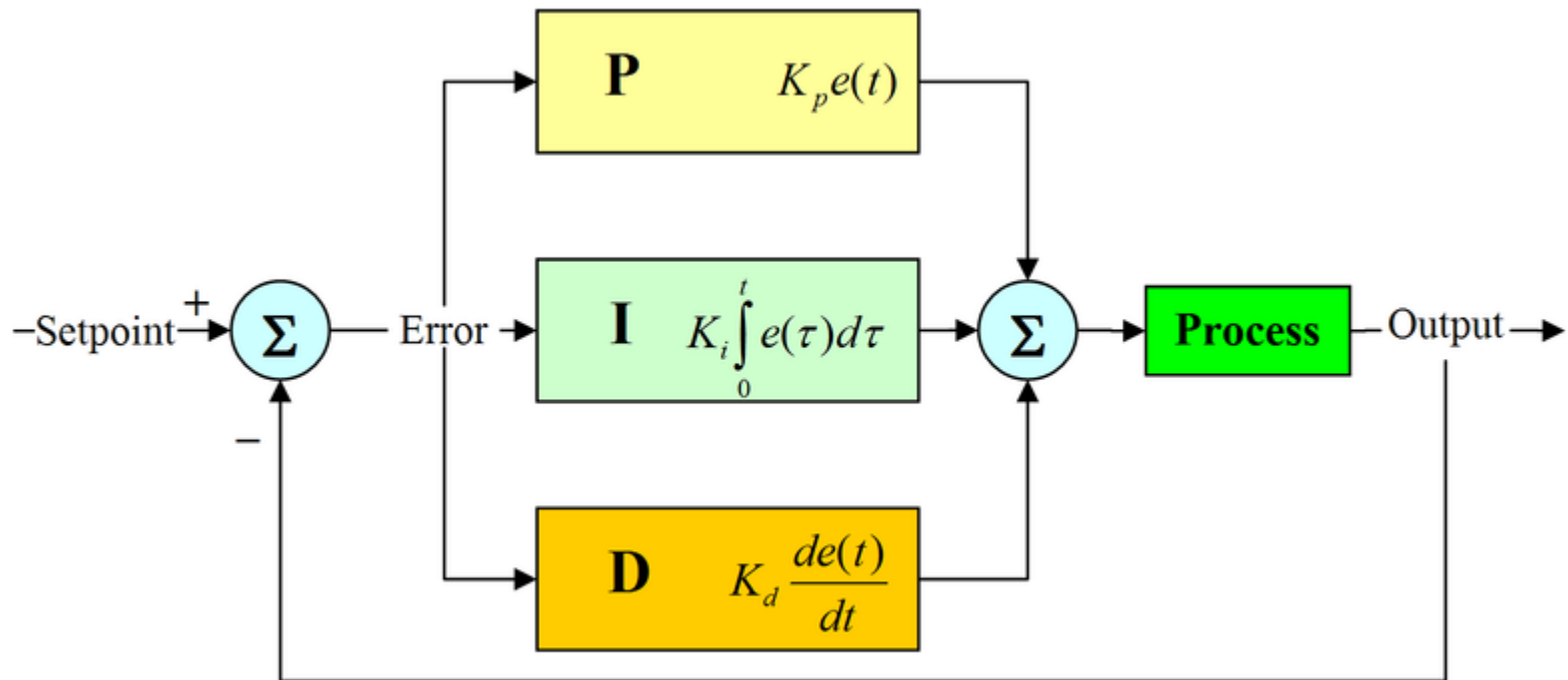
void A_motorStop() {
  digitalWrite(A_MotorDir1, LOW);
  digitalWrite(A_MotorDir2, LOW);
  digitalWrite(A_MotorEnable, LOW);
}

void A_motorF(){
  digitalWrite(A_MotorDir1,LOW);
  digitalWrite(A_MotorDir2,HIGH);
}

void A_motorR() {
  digitalWrite(A_MotorDir1, HIGH);
  digitalWrite(A_MotorDir2, LOW);
}
```

PID szabályozó

konyhanyelven



PID szabályozó

konyhanyelven

- P: Proporciónális tag – pillanatnyi hiba
 - $\text{ActualError} = \text{desiredValue} - \text{actualValue}$
 - $K_p * \text{ActualError}$
- I: Integráló tag – „időbeli” hiba
 - $K_i * \text{SumError}$
- D: Derivatív tag – hiba változási sebessége
 - $K_d * (\text{ActualError} - \text{PreviousError})$
- PID szabályozó jel: P + I + D tagok összege=
`controlSignal =`
 - `PID.P * error.ActualError +`
 - `PID.I * error.SumError +`
 - `PID.D * (error.ActualError - error.PreviousError);`

Új struktúrák

```
Struct errorContainer
{
    float ActualError;
    float PreviousError;
    float SumError;
};
```

```
struct PIDChannel
{
    float P;
    float I;
    float D;
};
```

```
errorContainer error;
PIDChannel PID;
```

```
float getControlSignal(float actualValue, float desiredValue)
{
    float controlSignal;

    error.ActualError=desiredValue-actualValue;
    error.SumError += error.ActualError;
    error.SumError = limit(error.SumError,-10, 10);

    controlSignal = PID.P * error.ActualError +
        PID.I * error.SumError +
        PID.D * (error.ActualError - error.PreviousError);
    error.PreviousError = error.ActualError;
    return controlSignal;
}
```


Slew Rate

