

Benedek Szakali
Anna Eszter Nyíri
Dániel Mihalik

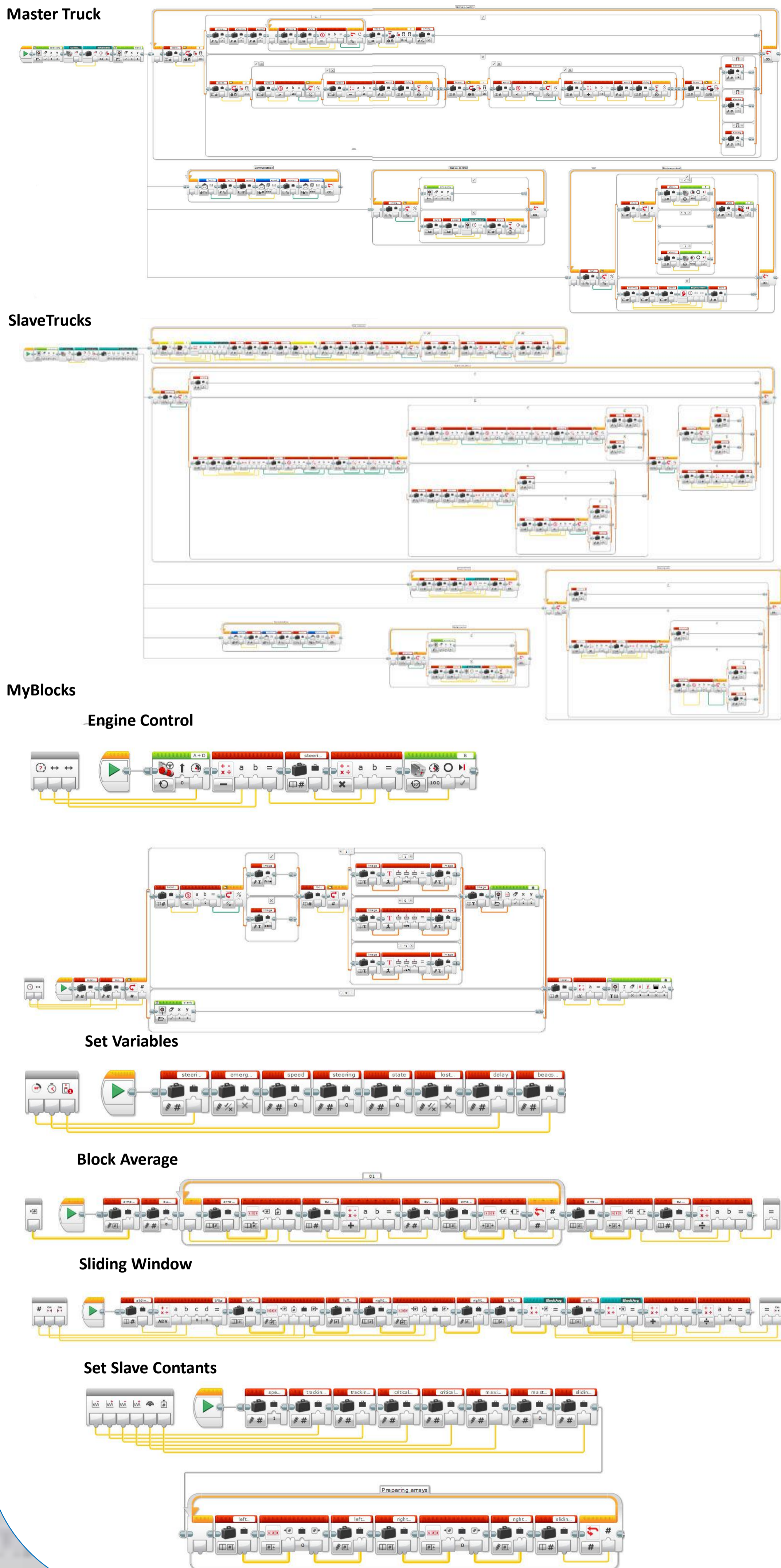
HUNGARY



Swarm Truck System



Swarm Truck System (EV3-G node based programming language)

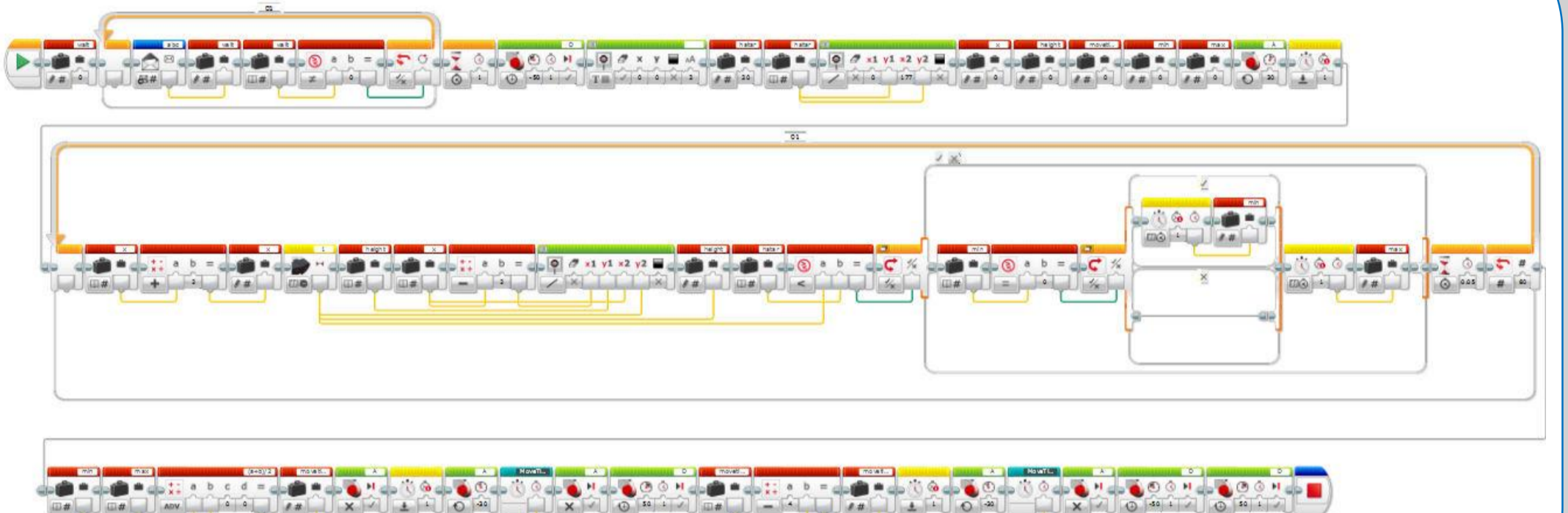


- Realistic drive and steering system
- Infrared and ultrasonic relative position determination
- Network communication
- Intelligent tracking system

Crane



Crane (EV3-G node based programming language)

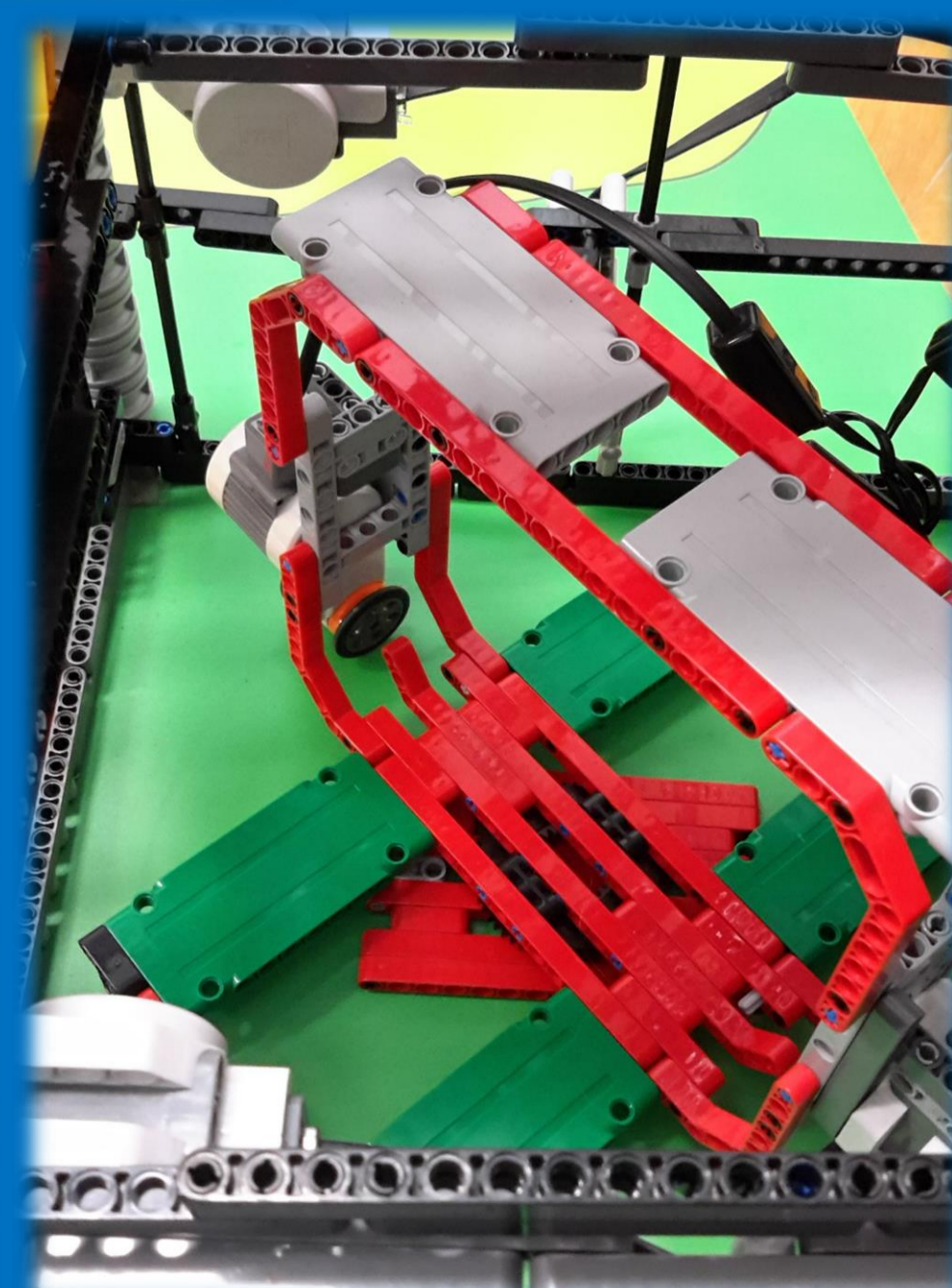
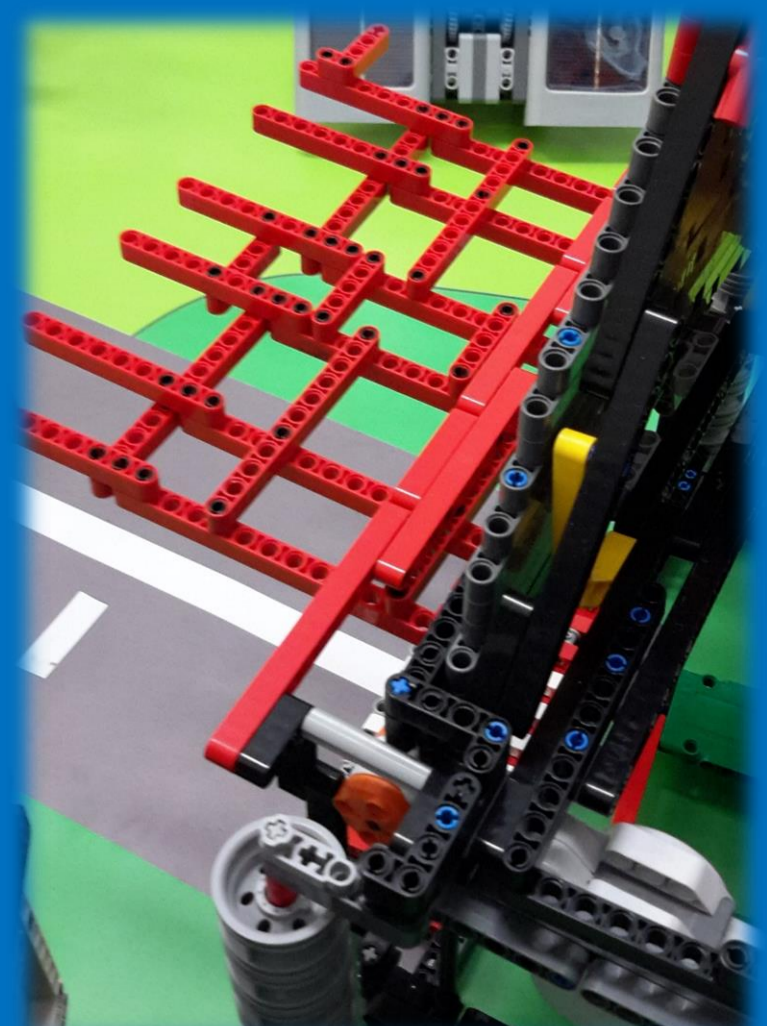


- Self-driving truck
- Automatic deposition system
- Infrared network communication

Transport



- Self-driving car
- Automatic garage door
- Automatic traffic management
- Car positioning (rotating) system
- Bluetooth communication



Self-driving car

```
int threshold = 40, speedHigh=40, speedLow=25;
int temp = 0;

void BT_fel_csato(int x, string nev){
  CommBTConnectionType args;
  args.Name = nev;
  args.ConnectionSlot = x;
  args.Action = 1;
  SysCommBTConnection(args);
  while(BluetoothStatus(x)!=0){
  }
}

void follow(){
  while(Sensor(S1) < threshold){
    if(Sensor(S2) > threshold){
      OnFwd(OUT_B, speedHigh);
      OnFwd(OUT_C, speedLow);
    } else {
      OnFwd(OUT_C, speedHigh);
      OnFwd(OUT_B, speedLow);
    }
  }
  NumOut(0, 0, Sensor(S1), true);
  Off(OUT_BC);
}

void intoGarage() { //Into The garage
  SendRemoteNumber(1,1,1); //Garage open
  Wait(2047);
  RotateMotor(OUT_BC,50,1300);
  OnFwd(OUT_B,-50);
  Wait(100);
  Off(OUT_BC);
  SendRemoteNumber(1,1,1); //Garage close
  Wait(10000);
}

void waitForTheLamp(int ms) {
  temp = 0;
  while(temp == 0){
    ReceiveRemoteNumber(1,true,temp);
    RotateMotor(OUT_BC,50,ms);
  }
}

task main(){
  SetSensorColorRed(IN_2);
  SetSensorColorRed(IN_1);

  BT_fel_csato(1,"garazs");
  PlayTone(700, 0.1);
  Wait(500);
  BT_fel_csato(2,"lampa");
  PlayTone(700, 0.1);
  Wait(500);

  while(true){
    //Garage open
    SendRemoteNumber(1,1,1);
    Wait(2000);
    OnFwdSync(OUT_BC, 50, 0.5);
    Wait(2000);
    Off(OUT_BC);
    OnFwd(OUT_C, 40);
    OnFwd(OUT_B, 30);
    while(Sensor(IN_2) < threshold){
      Off(OUT_BC);
    }
    follow();

    //Garage close
    SendRemoteNumber(1,1,1);
    //Wait for the lamp (into front)
    WaitForTheLamp(117);
    follow();

    //Turn back
    OnFwd(OUT_B, 50);
    OnFwd(OUT_C, -50);
    Wait(500);
    while(Sensor(IN_2) < threshold){
      Wait(200);
      while(Sensor(IN_2) > threshold){
        Wait(100);
        Off(OUT_BC);
      }
      follow();

      //Wait for the lamp (into back)
      WaitForTheLamp(117);
      follow();
      intoGarage();
    }
  }
}
```



Garage

```
bool garageOpen;
bool rotated = false;
void ChangeGarageState(I
  if(garageOpen){
    RotateMotor(OUT_AC, -50, 100);
  } else {
    RotateMotor(OUT_AC, 50, 110);
    OnFwd(OUT_AC,10);
  }
  garageOpen = !garageOpen;
}

void Rotate(){
  if(rotated){
    int val = -1;
    OnFwd(OUT_B, 50);
    Wait(1500);
    while (Sensor(IN_4)<40);
    Off(OUT_B);
  } else {
    OnFwd(OUT_B, -50);
    Wait(1500);
    while (Sensor(IN_4)<40);
    Off(OUT_B);
    rotated = !rotated;
  }
}

task main(){
  int val = -1;
  SetSensorColorRed(IN_4);
  while(true){
    ReceiveRemoteNumber(1,true,val);
    if(val != -1){
      ChangeGarageState();
    } else if(val == 2){
      Rotate();
    }
    val = -1;
  }
}
```