

ДОДАТОК А**Лістинг програми для керування автономного мобільного робота**

```
#include <Servo.h>
#include <RtcDS1302.h>
#include "RobotFunctions.h"
#include "PinConfig.h"

Servo horizontal;
int servoh = 180;
int servohLimitHigh = 175;
int servohLimitLow = 5;

Servo vertical;
int servov = 45;
int servovLimitHigh = 100;
int servovLimitLow = 1;

ThreeWire myWire(DAT_TIME, CLK_TIME, RST_TIME);
RtcDS1302<ThreeWire> Rtc(myWire);
bool isMorning = false;
int currentHour = 0;

void init_GPIO()
{
    horizontal.attach(1);
    vertical.attach(0);
    horizontal.write(180);
    vertical.write(45);
```

```
pinMode(RightMotorDirPin1, OUTPUT);
pinMode(RightMotorDirPin2, OUTPUT);
pinMode(speedPinL, OUTPUT);

pinMode(LeftMotorDirPin1, OUTPUT);
pinMode(LeftMotorDirPin2, OUTPUT);
pinMode(speedPinR, OUTPUT);
pinMode(RightMotorDirPin1B, OUTPUT);
pinMode(RightMotorDirPin2B, OUTPUT);
pinMode(speedPinLB, OUTPUT);

pinMode(LeftMotorDirPin1B, OUTPUT);
pinMode(LeftMotorDirPin2B, OUTPUT);
pinMode(speedPinRB, OUTPUT);

Rtc.Begin();

RtcDateTime currentTime = RtcDateTime(__DATE__ , __TIME__);
Rtc.SetDateTime(currentTime);
currentHour = Rtc.GetDateTime().Hour();
}

void setup()
{
  init_GPIO();
}

void loop(){
currentHour = Rtc.GetDateTime().Hour();
```

```
isMorning = (currentHour >= 6 && currentHour < 18);
```

```
RtcDateTime now = Rtc.GetDateTime();
```

```
int lt = analogRead(ldrLt);
```

```
int rt = analogRead(ldrRt);
```

```
int ld = analogRead(ldrLd);
```

```
int rd = analogRead(ldrRd);
```

```
int dtime = 10;
```

```
int tol = 90;
```

```
int avt = (lt + rt) / 2;
```

```
int avd = (ld + rd) / 2;
```

```
int avl = (lt + ld) / 2;
```

```
int avr = (rt + rd) / 2;
```

```
int dvert = avt - avd;
```

```
int dhoriz = avl - avr;
```

```
if (isMorning) {
```

```
    if (-1*tol > dvert || dvert > tol)
```

```
    {
```

```
        if (avt > avd)
```

```
        {
```

```
            servov = ++servov;
```

```
            if (servov > servovLimitHigh)
```

```
            {servov = servovLimitHigh;}
```

```
        }
```

```
        else if (avt < avd)
```

```
        {servov = --servov;
```

```
if (servov < servovLimitLow)
{ servov = servovLimitLow;}
}
vertical.write(servov);
}

if (-1*tol > dhoriz || dhoriz > tol)
{
if (avl > avr)
{
servoh = --servoh;
if (servoh < servohLimitLow)
{
servoh = servohLimitLow;
}
}
else if (avl < avr)
{
servoh = ++servoh;
if (servoh > servohLimitHigh)
{
servoh = servohLimitHigh;
}
}
else if (avl == avr)
{

stop_Stop();
```

```
delay(5000);
}
horizontal.write(servoh);
}

if (avt < avd) {
    go_advance(TURN_SPEED);
} else if (avl < avr) {
    right_turn(TURN_SPEED);
} else if (avl > avr) {
    left_turn(TURN_SPEED);
} else if (avt > avd) {
    go_back(TURN_SPEED);
} else {
    stop_Stop();
}
} else {
    vertical.write(45);
    horizontal.write(180);
    stop_Stop();
}

delay(dtime);
}
```

ДОДАТОК Б**Демонстраційний матеріал у вигляді презентацій**

