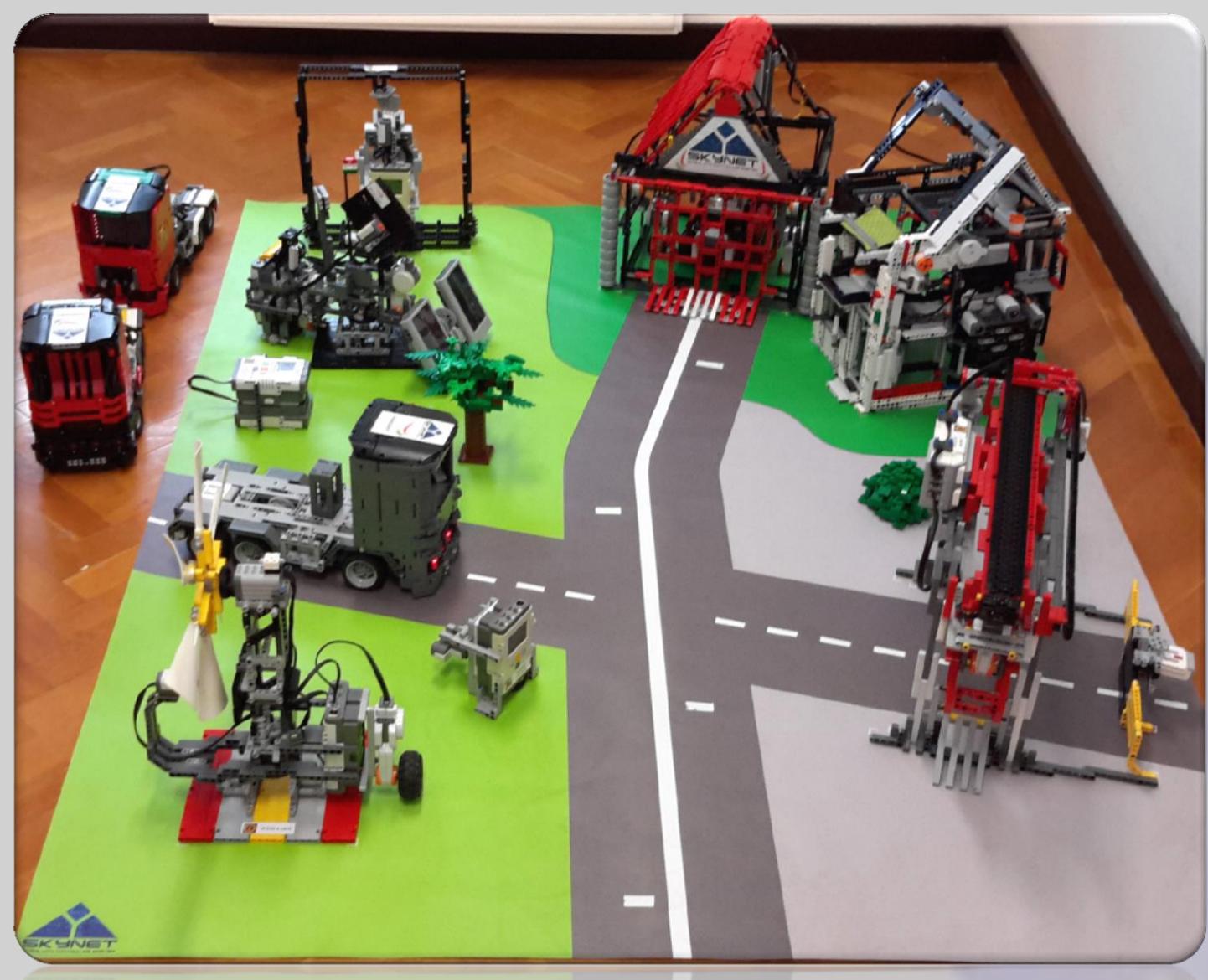# SkyNet City

Global City Controlling System

Benedek Szakali
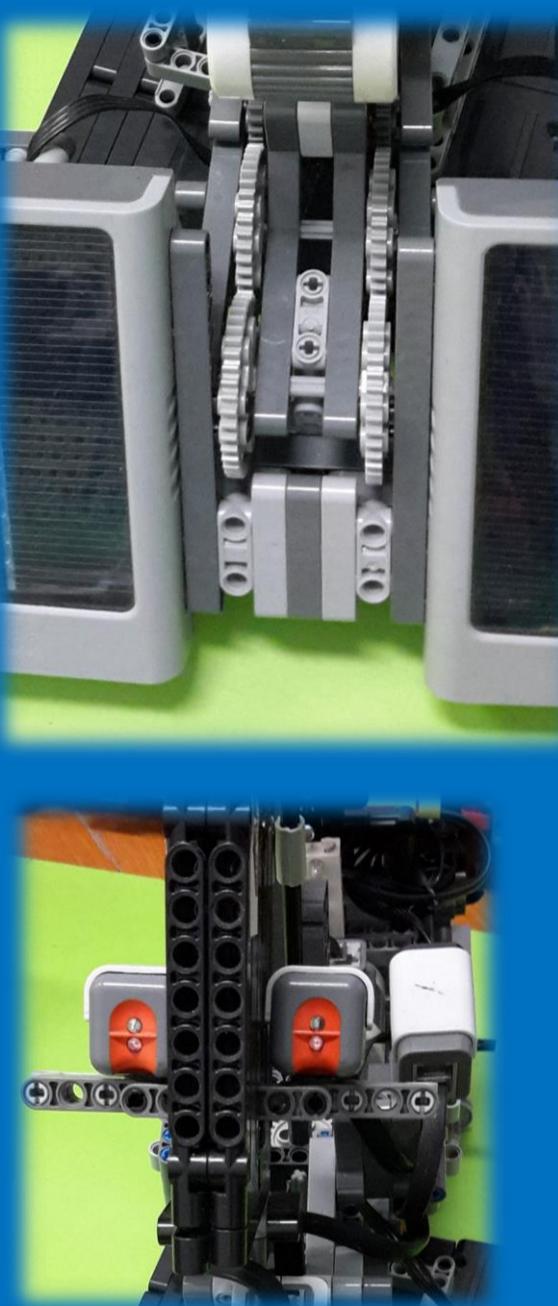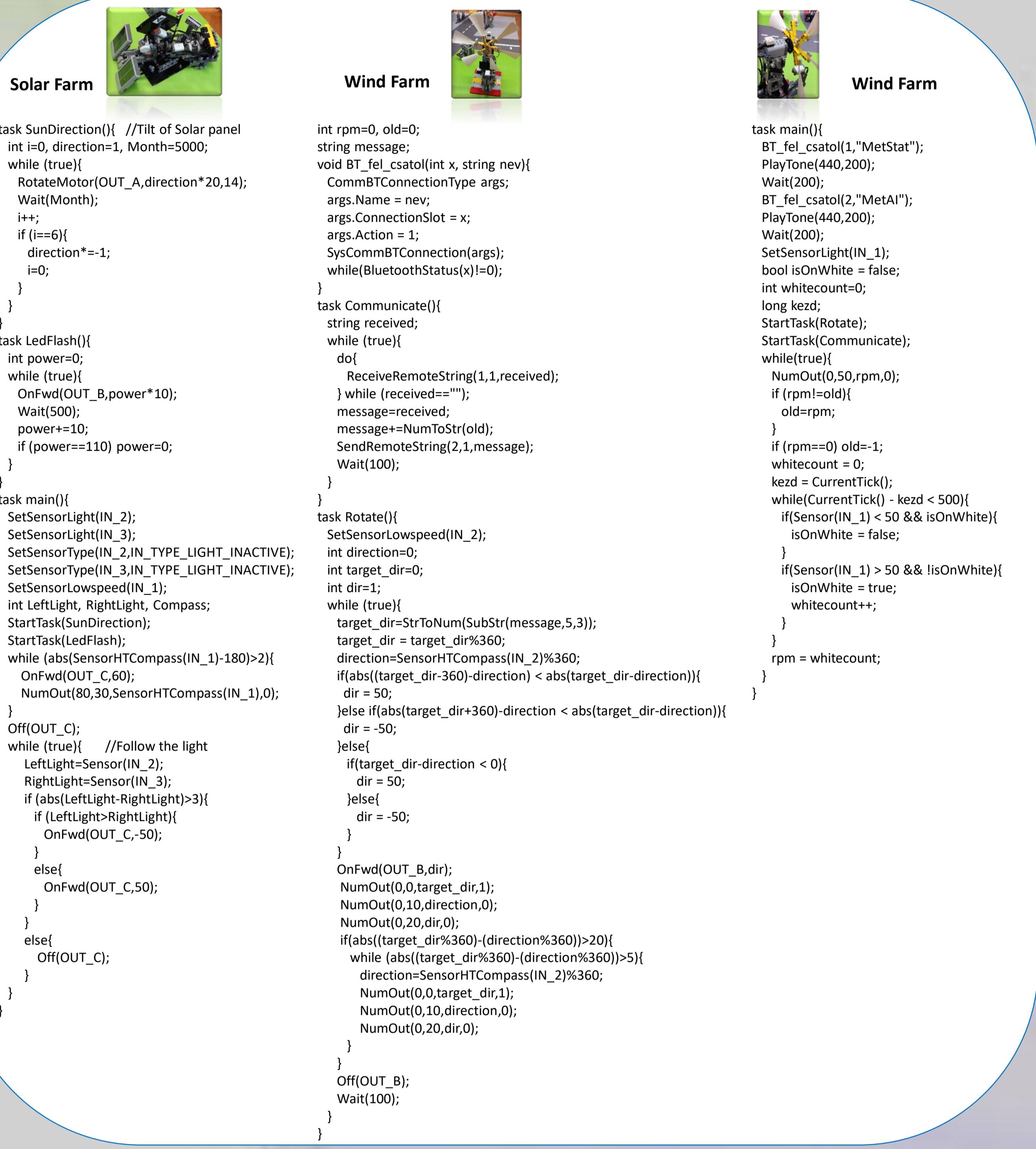**A**nna Eszter Nyíri
Dániel Mihalik

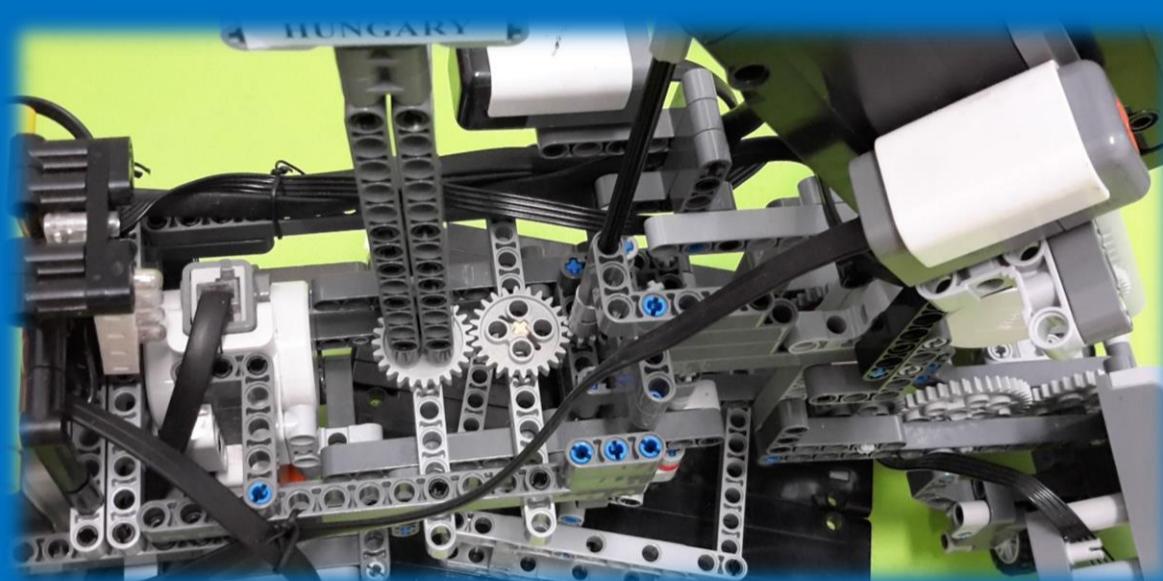HUNGARY

SKYNET
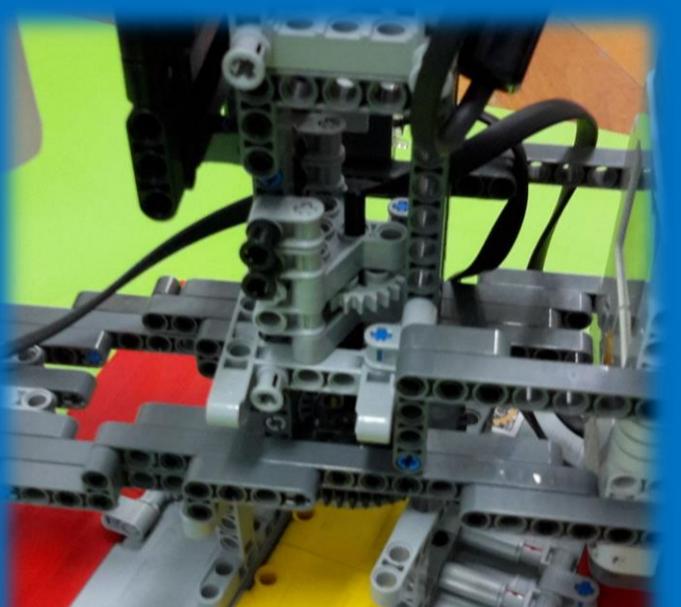GLOBAL CITY CONTROLLING SYSTEM

Renewable Energy

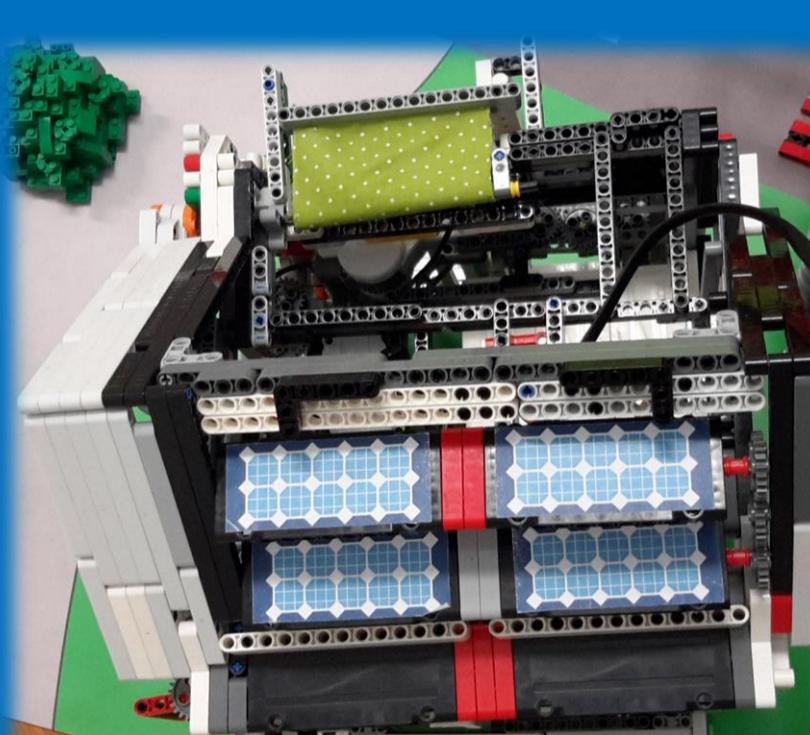### Solar Farm

Panels turning towards the brightest point automatically.

Determining the direction with compass sensors.
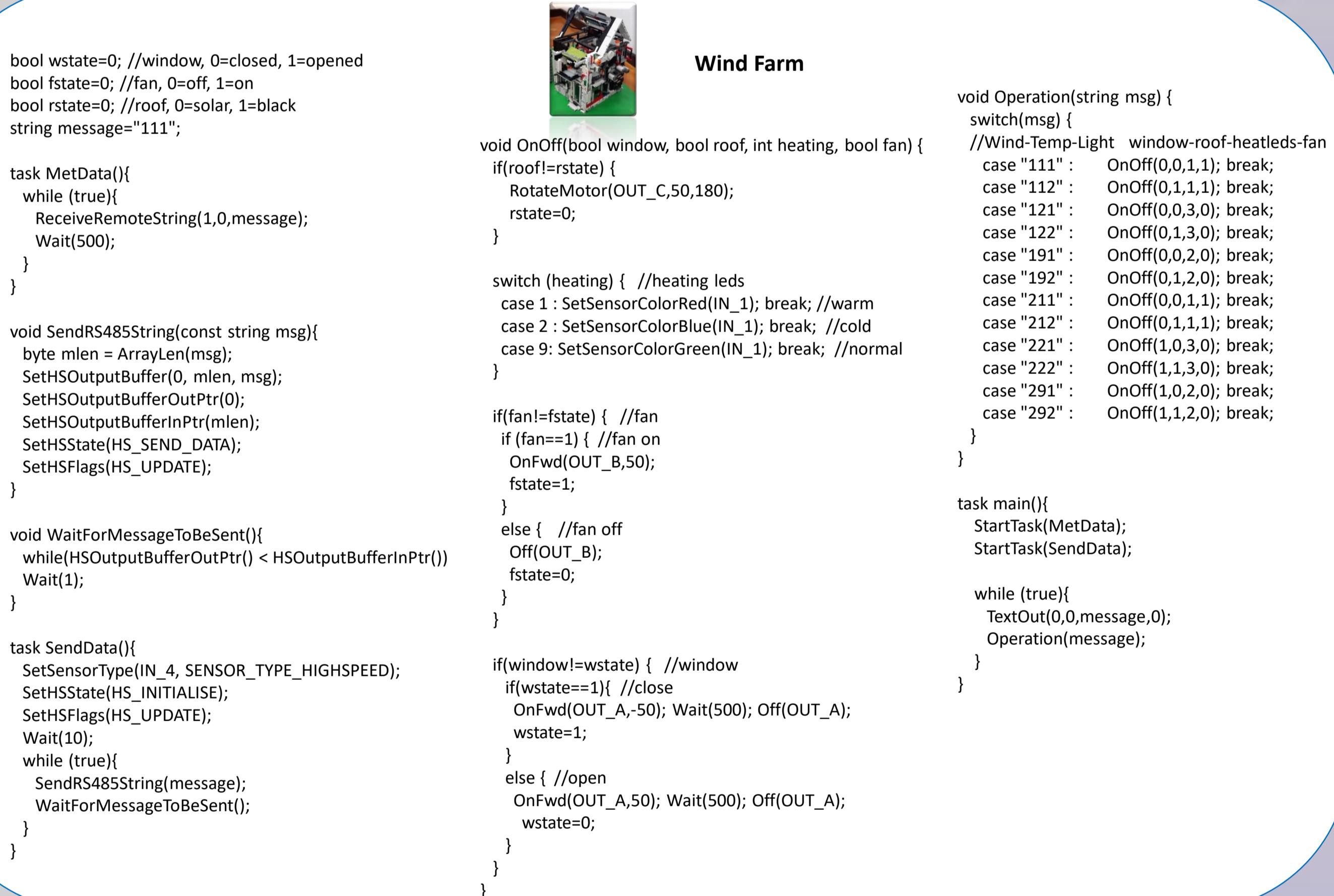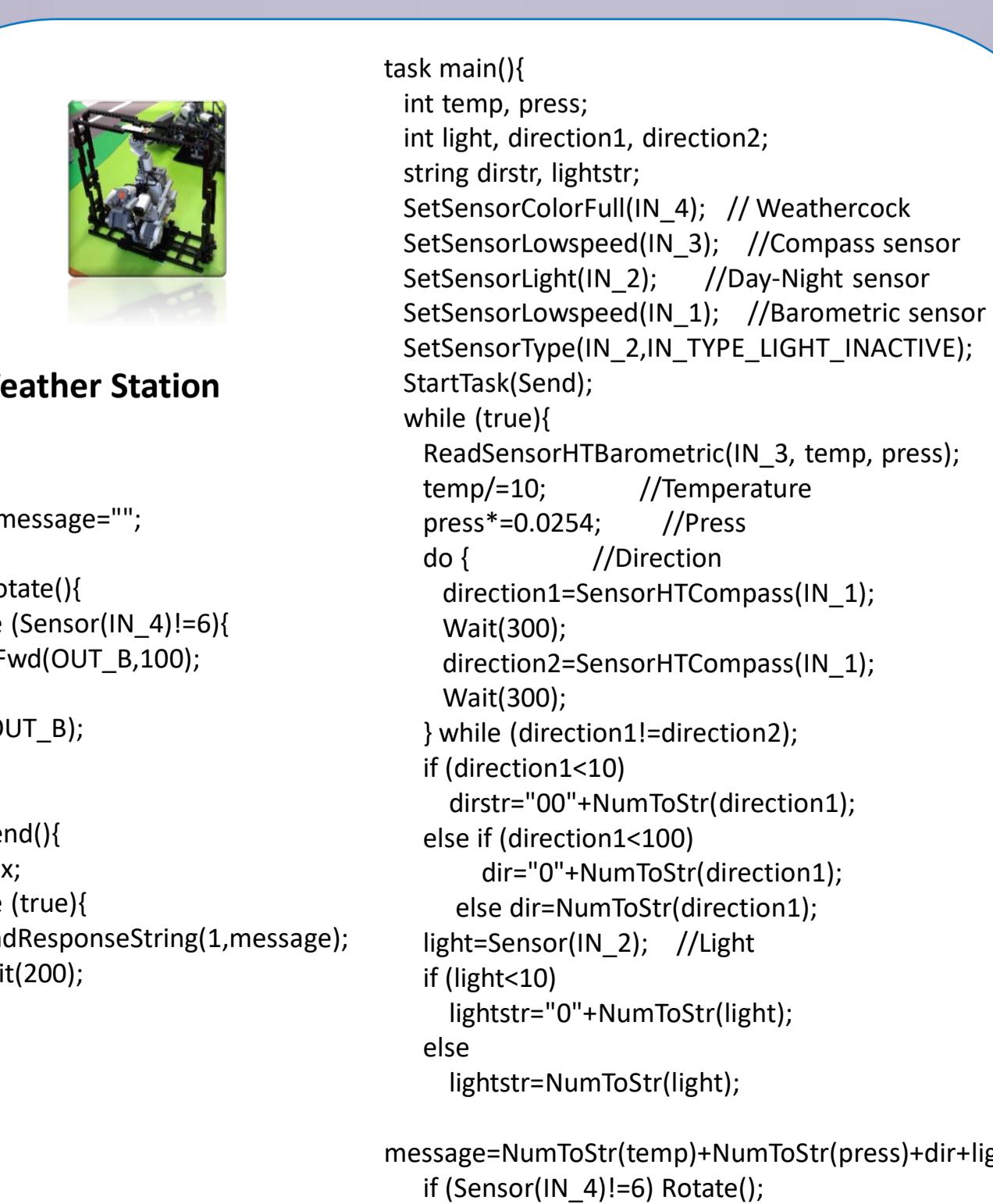
Wind Farm

Windmill turning towards the wind automatically.

Determining rotation speed with color sensors.

Solar Farm

```
task SunDirection(){   //Tilt of Solar panel
  int i=0, direction=1, Month=5000;
  while (true){
    RotateMotor(OUT_A,direction*20,14);
    Wait(Month);
    i++;
    if (i==6){
      direction*=-1;
      i=0;
    }
  }
}
task LedFlash(){
  int power=0;
  while (true){
    OnFwd(OUT_B,power*10);
    Wait(500);
    power+=10;
    if (power==110) power=0;
  }
}
task main(){
  SetSensorLight(IN_2);
  SetSensorLight(IN_3);
  SetSensorType(IN_2,IN_TYPE_LIGHT_INACTIVE);
  SetSensorType(IN_3,IN_TYPE_LIGHT_INACTIVE);
  SetSensorLowspeed(IN_1);
  int LeftLight, RightLight, Compass;
  StartTask(SunDirection);
  StartTask(LedFlash);
  while (abs(SensorHTCompass(IN_1)-180)>2){
    OnFwd(OUT_C,60);
    NumOut(80,30,SensorHTCompass(IN_1),0);
  }
  Off(OUT_C);
  while (true){   //Follow the light
    LeftLight=Sensor(IN_2);
    RightLight=Sensor(IN_3);
    if (abs(LeftLight-RightLight)>3){
      if (LeftLight>RightLight){
        OnFwd(OUT_C,-50);
      }
      else{
        OnFwd(OUT_C,50);
      }
    }
    else{
      Off(OUT_C);
    }
  }
}
```

Wind Farm

```
int rpm=0, old=0;
string message;
void BT_fel_csatol(int x, string nev){
  CommBTConnectionType args;
  args.Name = nev;
  args.ConnectionSlot = x;
  args.Action = 1;
  SysCommBTConnection(args);
  while(BluetoothStatus(x)!=0);
}
task Communicate(){
  string received;
  while (true){
    do{
      ReceiveRemoteString(1,1,received);
    } while (received=="");
    message=received;
    message+=NumToStr(old);
    SendRemoteString(2,1,message);
    Wait(100);
  }
}
task Rotate(){
  SetSensorLowspeed(IN_2);
  int direction=0;
  int target_dir=0;
  int dir=1;
  while (true){
    target_dir=StrToNum(SubStr(message,5,3));
    target_dir = target_dir%360;
    direction=SensorHTCompass(IN_2)%360;
    if(abs((target_dir-360)-direction) < abs(target_dir-direction)){
      dir = 50;
    }else if(abs((target_dir+360)-direction) < abs(target_dir-direction)){
      dir = -50;
    }else{
      if(target_dir-direction < 0){
        dir = 50;
      }else{
        dir = -50;
      }
    }
    OnFwd(OUT_B,dir);
    NumOut(0,0,target_dir,1);
    NumOut(0,10,direction,0);
    NumOut(0,20,dir,0);
    if(abs((target_dir%360)-(direction%360))>20){
      while (abs((target_dir%360)-(direction%360))>5){
        direction=SensorHTCompass(IN_2)%360;
        NumOut(0,0,target_dir,1);
        NumOut(0,10,direction,0);
        NumOut(0,20,dir,0);
      }
    }
    Off(OUT_B);
    Wait(100);
  }
}
```

Wind Farm

```
task main(){
  BT_fel_csatol(1,"MetStat");
  PlayTone(440,200);
  Wait(200);
  BT_fel_csatol(2,"MetAI");
  PlayTone(440,200);
  Wait(200);
  SetSensorLight(IN_1);
  bool isOnWhite = false;
  int whitecount=0;
  long kezd;
  StartTask(Rotate);
  StartTask(Communicate);
  while(true){
    NumOut(0,50,rpm,0);
    if (rpm!=old){
      old=rpm;
    }
    if (rpm==0) old-=1;
    whitecount = 0;
    kezd = CurrentTick();
    while(CurrentTick() - kezd < 500){
      if(Sensor(IN_1) < 50 && isOnWhite){
        isOnWhite = false;
      }
      if(Sensor(IN_1) > 50 && !isOnWhite){
        isOnWhite = true;
        whitecount++;
      }
    }
    rpm = whitecount;
  }
}
```

Smart Home

Doors, windows, lights, roof structure reacting to weather data.


```
bool wstate=0;  //window, 0=closed, 1=opened
bool fstate=0;  //fan, 0=off, 1=on
bool rstate=0;  //roof, 0=solar, 1=black
string message="111";

task MetData(){
  while (true){
    ReceiveRemoteString(1,0,message);
    Wait(500);
  }
}

void SendRS485String(const string msg){
  byte mlen = ArrayLen(msg);
  SetHSOutputBuffer(0, mlen, msg);
  SetHSOutputBufferOutPtr(0);
  SetHSOutputBufferInPtr(mlen);
  SetHSState(HS_SEND_DATA);
  SetHSFlags(HS_UPDATE);
}

void WaitForMessageToBeSent(){
  while(HSOutputBufferOutPtr() < HSOutputBufferInPtr())
  Wait(1);
}

task SendData(){
  SetSensorType(IN_4, SENSOR_TYPE_HIGHSPEED);
  SetHSState(HS_INITIALISE);
  SetHSFlags(HS_UPDATE);
  Wait(10);
  while (true){
    SendRS485String(message);
    WaitForMessageToBeSent();
  }
}
```

Wind Farm

```
void OnOff(bool window, bool roof, int heating, bool fan) {
  if(roof!=rstate) {
    RotateMotor(OUT_C,50,180);
    rstate=0;
  }

  switch (heating) {   //heating leds
    case 1 : SetSensorColorRed(IN_1); break;   //warm
    case 2 : SetSensorColorBlue(IN_1); break;   //cold
    case 9 : SetSensorColorGreen(IN_1); break;   //normal
  }

  if(fan!=fstate) {   //fan
    if (fan==1) {   //fan on
      OnFwd(OUT_B,50);
      fstate=1;
    }
    else {   //fan off
      Off(OUT_B);
      fstate=0;
    }
  }

  if(window!=wstate) {   //window
    if(wstate==1){   //close
      OnFwd(OUT_A,-50); Wait(500); Off(OUT_A);
      wstate=1;
    }
    else {   //open
      OnFwd(OUT_A,50); Wait(500); Off(OUT_A);
      wstate=0;
    }
  }
}
```

```
void Operation(string msg) {
  switch(msg) {
    //Wind-Temp-Light  window-roof-heatleds-fan
    case "111" :  OnOff(0,0,1,1); break;
    case "112" :  OnOff(0,0,1,1); break;
    case "121" :  OnOff(0,0,3,0); break;
    case "122" :  OnOff(0,1,3,0); break;
    case "191" :  OnOff(0,0,2,0); break;
    case "192" :  OnOff(0,0,2,0); break;
    case "211" :  OnOff(0,0,1,1); break;
    case "212" :  OnOff(0,1,1,1); break;
    case "221" :  OnOff(1,0,3,0); break;
    case "222" :  OnOff(1,1,3,0); break;
    case "291" :  OnOff(1,0,2,0); break;
    case "292" :  OnOff(1,1,2,0); break;
  }
}

task main(){
  StartTask(MetData);
  StartTask(SendData);

  while (true){
    TextOut(0,0,message,0);
    Operation(message);
  }
}
```

Data Transfer Center

Data Transfer Center
Dual NXT


```
Light: Night (13)
Wind: North
Wind pow.:80 rpm
Press: 759.23Hgmm
Temp.: 26.6 ℃
```

```
string message="32759123560";
int rpm=0;
string TextLight(int light){
  string text_light;
  if (light<20) text_light="Night";
  else if (light<50) text_light="Cloudy";
  else text_light="Sunny";
  return text_light;
}

string TextDirection(int direction){
  string text_direction;
  if (direction<23) text_direction= "South   ";
  else if (direction<68) text_direction= "SouthWest";
  else if (direction<113) text_direction="West   ";
  else if (direction<158) text_direction="NorthWest";
    else if (direction<203) text_direction="North   ";
      else if (direction<248) text_direction="NorthEast";
        else if (direction<293) text_direction="East   ";
          else if (direction<338) text_direction= "SouthEast";
            else text_direction="South   ";
  if (rpm==0) text_direction="No wind ";
  return text_direction;
}

void RefreshScreen(){
  int num;
  string answer;
  TextOut(0,0,"Temp.: ",1);   //temp
  TextOut(44,0,SubStr(message,0,2),0);
  CircleOut(75,5,2);
  TextOut(79,0,"C",0);
  TextOut(0,16,"Press: ",0);   //press
  TextOut(38,16,SubStr(message,2,3),0);
  TextOut(75,16,"Hgmm",0);
  TextOut(0,32,"Wind pow.: ",0);   //wind power
  rpm=StrToNum(SubStr(message,10,1));
  NumOut(60,32,rpm*20,0);
  TextOut(80,32,"rpm",0);
  TextOut(0,40,"Wind: ",0);   //TextOut(40,40,SubStr(message,11,3),0);
  num=StrToNum(SubStr(message,5,3));
  answer=TextDirection(num);
  TextOut(40,40,answer,0);
  TextOut(0,56,"Light: ",0);   //light
  num=StrToNum(SubStr(message,8,2));
  answer=TextLight(num);
  TextOut(38,56,answer,0);
  TextOut(75,56,"(",0);
  TextOut(80,56,SubStr(message,8,2),0);
  TextOut(92,56,")",0);
}
```

```
void SendRS485String(const string msg){
  byte mlen = ArrayLen(msg);
  SetHSOutputBuffer(0, mlen, msg);
  SetHSOutputBufferOutPtr(0);
  SetHSOutputBufferInPtr(mlen);
  SetHSState(HS_SEND_DATA);
  SetHSFlags(HS_UPDATE);   //send it
}
void WaitForMessageToBeSent(){
  while(HSOutputBufferOutPtr() < HSOutputBufferInPtr())
  Wait(1);
}
task SmartHomeData(){
  SetSensorType(IN_4, SENSOR_TYPE_HIGHSPEED);
  SetHSState(HS_INITIALISE);
  SetHSFlags(HS_UPDATE);
  Wait(10);
  string HomeData;
  float temp, light;
  while (true)  {
    if ((rpm*20)>=40) HomeData="1";
    else HomeData="2";
    temp=StrToNum(SubStr(message,0,2));
    if (temp>25) HomeData="1";
    if (temp<18) HomeData="2";
    if ((temp<=25) && (temp>=18))
      HomeData="9";
    light=StrToNum(SubStr(message,8,2));
    if (light<20) HomeData+="1";
    else HomeData+="2";
    SendRS485String(HomeData);
    WaitForMessageToBeSent();
  }
}
task WindData(){
  int data;
  while (true){
    do{
      ReceiveRemoteNumber(2,1,data);
      Wait(100);
    }while (data==0);
    if (data<0) rpm=0;
    else rpm=data;
  }
}
task RefreshData(){
  long begin;
  while (true){
    begin=CurrentTick();
    while (abs(CurrentTick()-begin)<1000);
    RefreshScreen();
  }
}

task main(){
  string received;
  StartTask(WindData);
  StartTask(RefreshData);
  StartTask(SmartHomeData);
  while (true){
    do{
      ReceiveRemoteString(1,1,received);
    } while (received=="");
    message=received;
    Wait(100);
  }
}
```

Weather Station

The station measures the temperature, air pressure, light in the surroundings and the wind direction. Data sent to the data transfer center.

Weather Station

```
string message="";

void Rotate(){
  while (Sensor(IN_4)!=6){
    OnFwd(OUT_B,100);
  }
  Off(OUT_B);
}

task Send(){
  float x;
  while (true){
    SendResponseString(1,message);
    Wait(200);
  }
}
```

```
task main(){
  int temp, press;
  int light, direction1, direction2;
  string dirstr, lightstr;
  SetSensorColorFull(IN_4);   // Weathercock
  SetSensorLowspeed(IN_3);   //Compass sensor
  SetSensorLight(IN_2);   //Day-Night sensor
  SetSensorLowspeed(IN_1);   //Barometric sensor
  SetSensorType(IN_2,IN_TYPE_LIGHT_INACTIVE);
  StartTask(Send);
  while (true){
    ReadSensorHTBarometric(IN_3, temp, press);
    temp/=10;   //Temperature
    press*=0.0254;   //Press
    do {   //Direction
      direction1=SensorHTCompass(IN_1);
      Wait(300);
      direction2=SensorHTCompass(IN_1);
      Wait(300);
    } while (direction1!=direction2);
    if (direction1<10)
      dirstr="00"+NumToStr(direction1);
    else if (direction1<100)
      dir="0"+NumToStr(direction1);
    else dir=NumToStr(direction1);
    light=Sensor(IN_2);   //Light
    if (light<10)
      lightstr="0"+NumToStr(light);
    else
      lightstr=NumToStr(light);

  message=NumToStr(temp)+NumToStr(press)+dir+lig;
  if (Sensor(IN_4)!=6) Rotate();
  }
}
```