

Rok. ak. 2021/2022
grupa dziekańska: 6AiSR2
termin realizowania laboratorium: czwartek 8:15

SPRAWOZDANIE Z LABORATORIUM OPROGRAMOWANIA SYSTEMÓW MIKROPROCESOROWYCH

Autorzy sprawozdania:

1. Elena Gricjuta 228391_____

2. Arkadiusz Jóźwiak 228401_____

imię

nazwisko

nr indeksu

Data wykonania sprawozdania: 30.05.2022

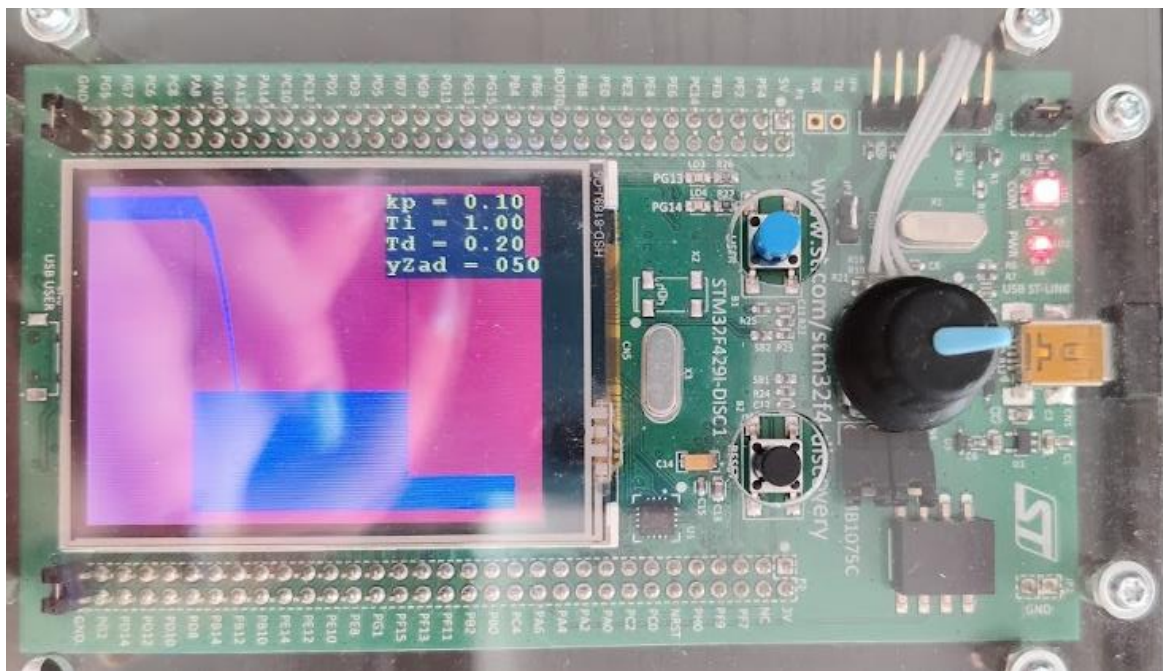
Spis treści

1.1. Charakterystyka oprogramowania	3
1.2. Schemat blokowy architektury oprogramowania	6
1.3. Kod źródłowy	7

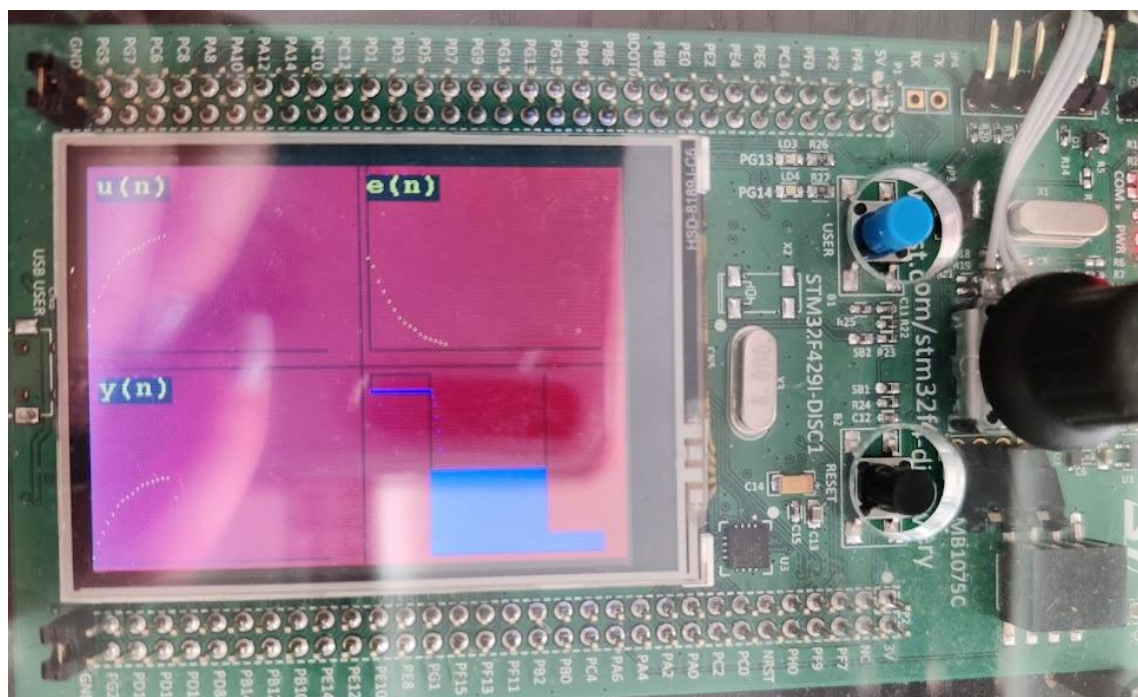
1. Portowanie symulatora na platformę STM32F429 z systemem FreeRTOS

1.1. Charakterystyka oprogramowania

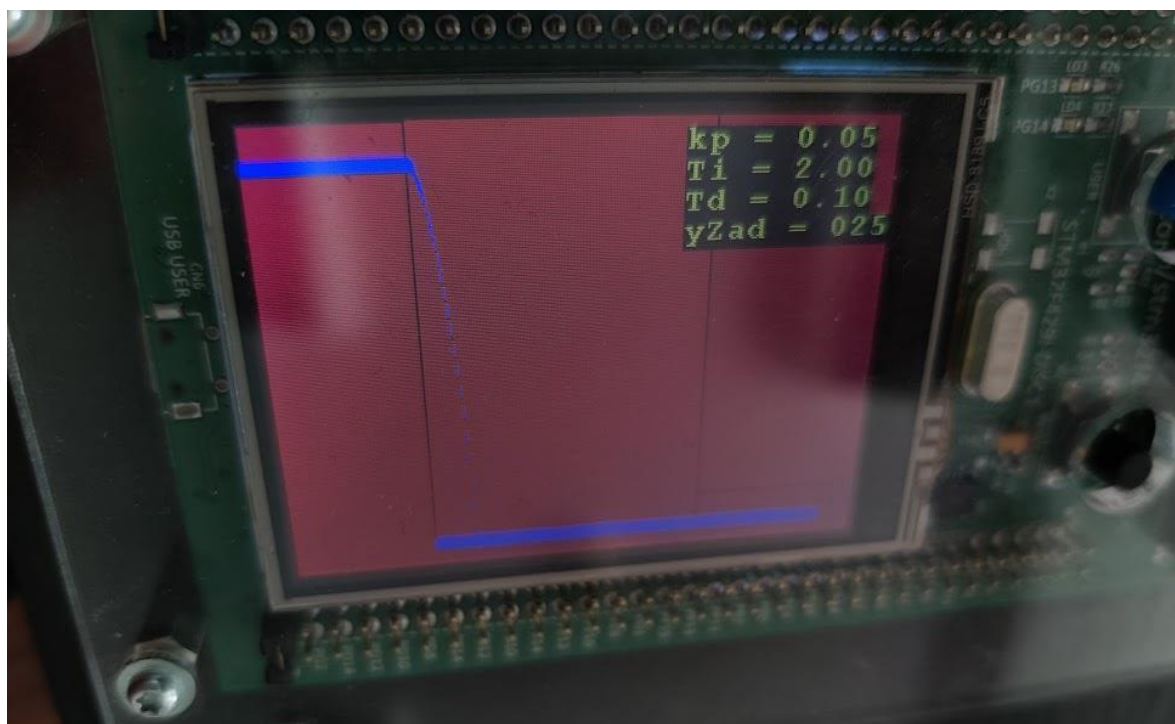
Celem danego zadania było zrealizowanie dokładnie takiego samego układu regulacji z regulatorem PID, który był zaprojektowany wcześniej, ale tym razem na platformie STM32F429 z systemem FreeRTOS. Powstały program realizuje dokładnie wszystkie założenia, co wcześniej. Powstało 6 ekranów: w tym wszystkie nieparzyste wyświetlały zbiornik oraz nastawy, natomiast parzyste wyświetlały zbiornik oraz powstałe wykresy.



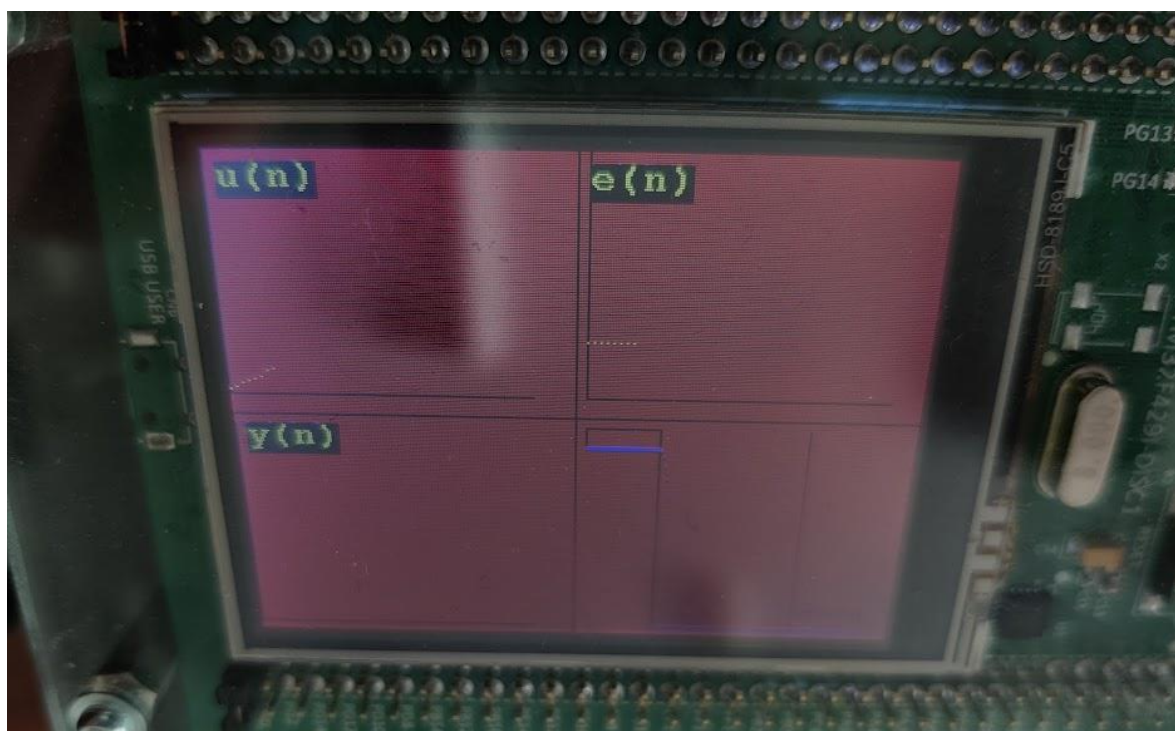
Rysunek 1 - Ekran 1



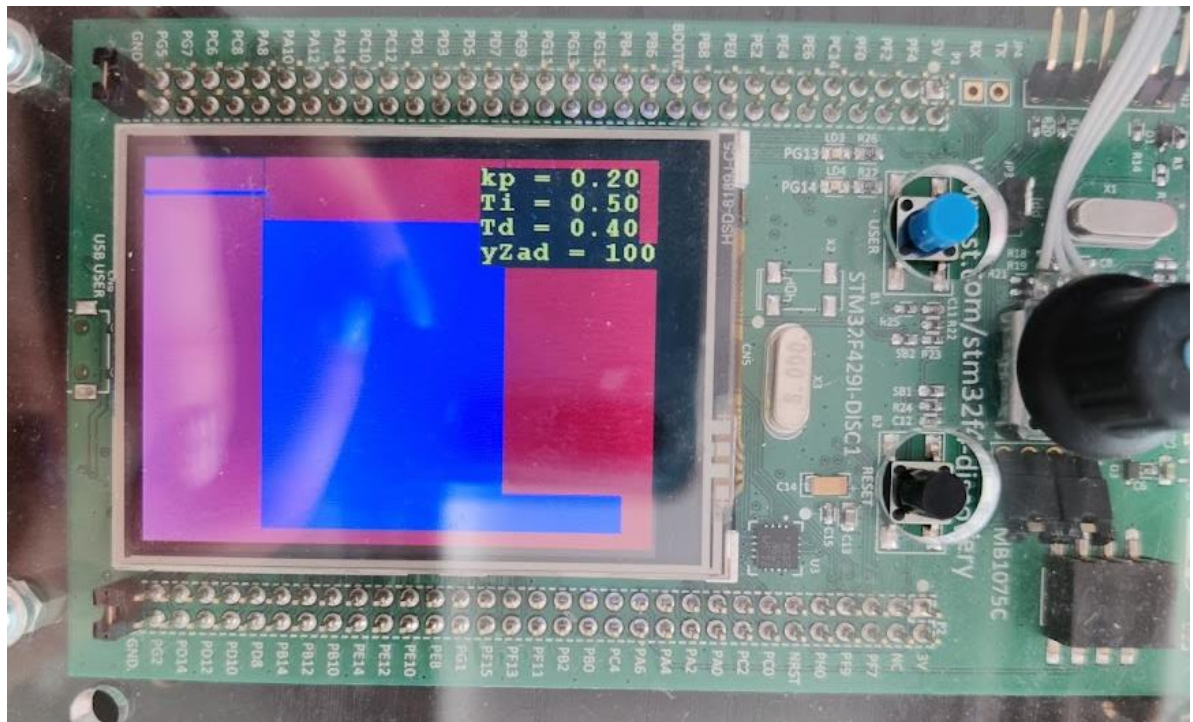
Rysunek 2 - Ekran 2



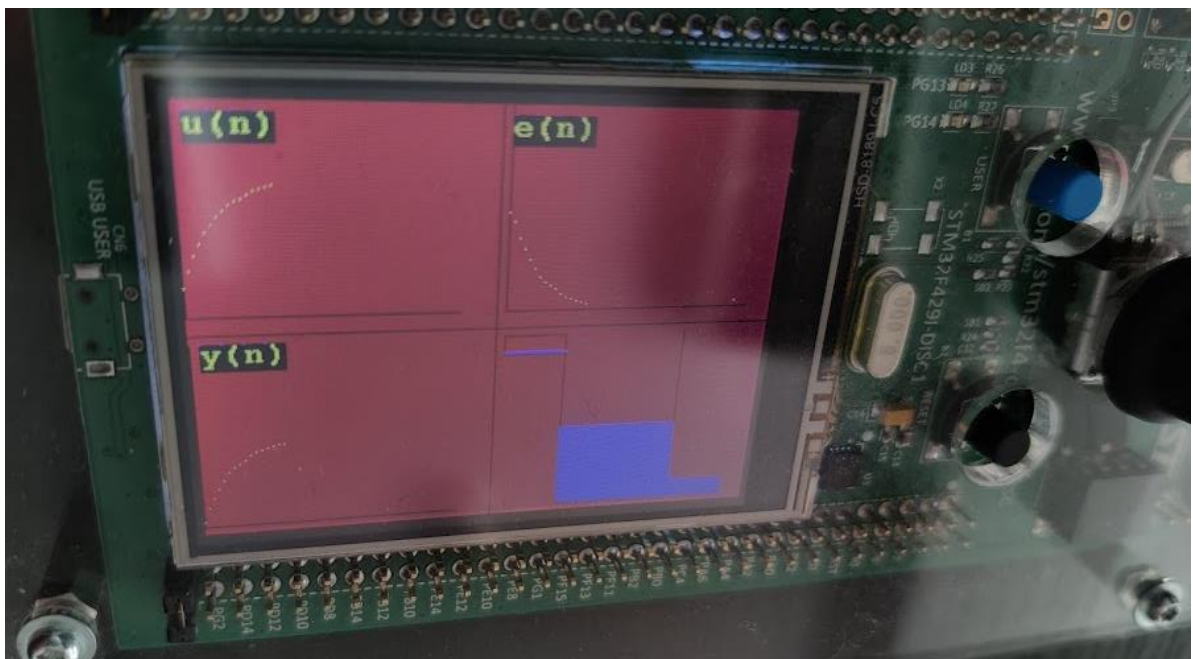
Rysunek 3 - Ekran 3



Rysunek 4 - Ekran 4

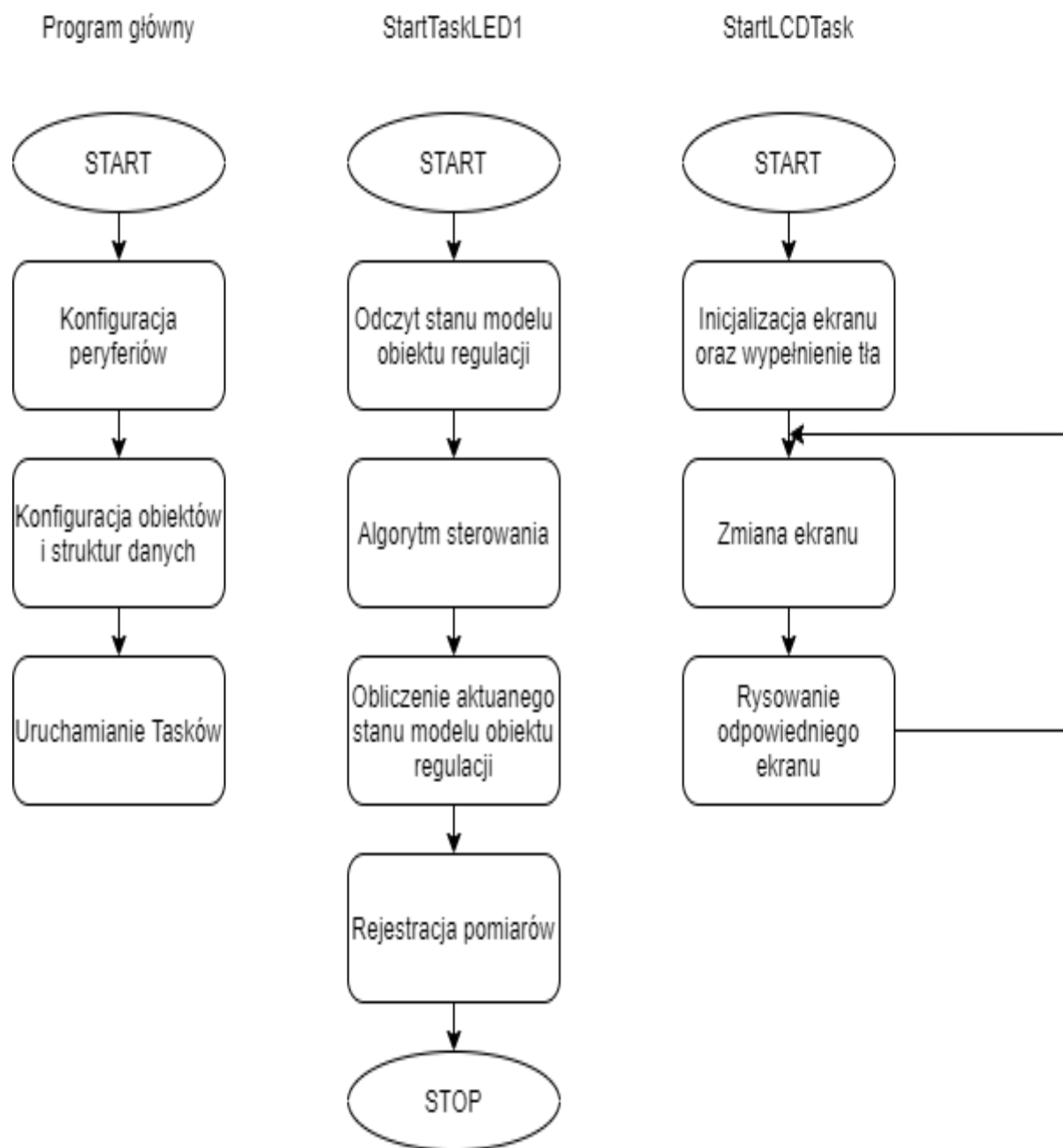


Rysunek 5 - Ekran 5



Rysunek 6 - Ekran 6

1.2. Schemat blokowy architektury oprogramowania



Rysunek 7 - Schemat blokowy

1.3. Kod źródłowy

StartLCDTask

```

void StartLCDTask(void *argument)
{
    /* USER CODE BEGIN 5 */
    TFTDisplay_ILI9341_Initialization(240, 320);
    TFTDisplay_ILI9341_FillScreen(TFT_COLOR_ILI9341_RED);
    TFTDisplay_ILI9341_SetRotation(1);
    for (;;)
    {
        TFTDisplay_ILI9341_SetTextColor(TFT_COLOR_ILI9341_YELLOW);
        if (GPIOA->IDR & 0x0001)
        {
            if (!block)
            {
                screenNumber++;
                if (screenNumber == 3 || screenNumber == 5 || screenNumber ==
7)
                {
                    yWy = 0;
                    yWyPrev = 0;
                    iv = 0;
                    PID_Calculate(0, 1);
                    clearArray(v_u, nv);
                    clearArray(v_e, nv);
                    clearArray(v_yWy, nv);
                    if (screenNumber == 7)
                        screenNumber = 1;
                }
                block = 1;
            }
            else
            {
                block = 0;
            }
            if (Tim != TimPrev)
            {
                switch (screenNumber)
                {
                    case 1:
                        WaterTank_Pipes(t_a[1], t_H[1], t_x0[0], t_y0[0], yWy,
yWyPrev, H_MAX, z, 0);

                        temp = kp;
                        ch_kp[5] = '0' + temp;
                        ch_kp[6] = '.';
                        temp = (temp - (int)temp) * 10;
                        ch_kp[7] = '0' + temp;
                        temp = (temp - (int)temp) * 10;
                        ch_kp[8] = '0' + temp;

                        temp = Ti;
                        ch_Ti[5] = '0' + temp;
                        ch_Ti[6] = '.';
                        temp = (temp - (int)temp) * 10;
                        ch_Ti[7] = '0' + temp;
                        temp = (temp - (int)temp) * 10;
                        ch_Ti[8] = '0' + temp;

                        temp = Td;
                        ch_Td[5] = '0' + temp;
                        ch_Td[6] = '.';
                        temp = (temp - (int)temp) * 10;
                        ch_Td[7] = '0' + temp;
                        temp = (temp - (int)temp) * 10;
                        ch_Td[8] = '0' + temp;

                        temp = yZad;
                        ch_yZad[7] = '0' + (int)temp / 100;
                        temp = temp - ((int)temp / 100) * 100;
                        ch_yZad[8] = '0' + (int)temp / 10;
                        temp = temp - ((int)temp / 10) * 10;
                        ch_yZad[9] = '0' + (int)temp;

                        TFTDisplay_ILI9341_String(210, 5, ch_kp);
                        TFTDisplay_ILI9341_String(210, 20, ch_Ti);
                        TFTDisplay_ILI9341_String(210, 35, ch_Td);
                        TFTDisplay_ILI9341_String(210, 50, ch_yZad);
                        break;

                    case 2:
                        FunctionRealTime(t_a[0], t_H[0], t_x0[0], t_y0[0], v_u, H_MAX,
0);
                        FunctionRealTime(t_a[0], t_H[0], t_x0[1], t_y0[0], v_e, H_MAX,
0);
                        FunctionRealTime(t_a[0], t_H[0], t_x0[0], t_y0[1], v_yWy,
H_MAX, 0);
                        WaterTank_Pipes(t_a[0], t_H[0], t_x0[1], t_y0[1], yWy,
yWyPrev, H_MAX, z, 0);
                        Cross(0);

                        TFTDisplay_ILI9341_String(165, 5, ch_e);
                        TFTDisplay_ILI9341_String(5, 125, ch_y);
                        TFTDisplay_ILI9341_String(5, 5, ch_u);

                        break;

                    case 3:
                        WaterTank_Pipes(t_a[1], t_H[1], t_x0[0], t_y0[0], yWy,
yWyPrev, H_MAX, z, 0);

                        temp = kp;
                        ch_kp[5] = '0' + temp;
                        ch_kp[6] = '.';
                        temp = (temp - (int)temp) * 10;
                        ch_kp[7] = '0' + temp;
                        temp = (temp - (int)temp) * 10;
                        ch_kp[8] = '0' + temp;

                        temp = Ti;
                        ch_Ti[5] = '0' + temp;
                        ch_Ti[6] = '.';
                        temp = (temp - (int)temp) * 10;
                        ch_Ti[7] = '0' + temp;
                        temp = (temp - (int)temp) * 10;
                        ch_Ti[8] = '0' + temp;

                        temp = Td;
                        ch_Td[5] = '0' + temp;
                        ch_Td[6] = '.';
                        temp = (temp - (int)temp) * 10;
                        ch_Td[7] = '0' + temp;
                        temp = (temp - (int)temp) * 10;
                        ch_Td[8] = '0' + temp;

                        temp = yZad;
                        ch_yZad[7] = '0' + (int)temp / 100;
                        temp = temp - ((int)temp / 100) * 100;
                        ch_yZad[8] = '0' + (int)temp / 10;
                        temp = temp - ((int)temp / 10) * 10;
                        ch_yZad[9] = '0' + (int)temp;

                        TFTDisplay_ILI9341_String(210, 5, ch_kp);
                        TFTDisplay_ILI9341_String(210, 20, ch_Ti);
                        TFTDisplay_ILI9341_String(210, 35, ch_Td);
                        TFTDisplay_ILI9341_String(210, 50, ch_yZad);
                        break;

                    case 4:
                        FunctionRealTime(t_a[0], t_H[0], t_x0[0], t_y0[0], v_u, H_MAX,
0);
                        FunctionRealTime(t_a[0], t_H[0], t_x0[1], t_y0[0], v_e, H_MAX,
0);
                        FunctionRealTime(t_a[0], t_H[0], t_x0[0], t_y0[1], v_yWy,
H_MAX, 0);
                        WaterTank_Pipes(t_a[0], t_H[0], t_x0[1], t_y0[1], yWy,
yWyPrev, H_MAX, z, 0);

```

```

Cross(0);

TFTDisplay_ILI9341_String(165, 5, ch_e);
TFTDisplay_ILI9341_String(5, 125, ch_y);
TFTDisplay_ILI9341_String(5, 5, ch_u);

break;

case 5:
    WaterTank_Pipes(t_a[1], t_H[1], t_x0[0], t_y0[0], yWy,
yWyPrev, H_MAX, z, 0);

    temp = kp;
    ch_kp[5] = '0' + temp;
    ch_kp[6] = '.';
    temp = (temp - (int)temp) * 10;
    ch_kp[7] = '0' + temp;
    temp = (temp - (int)temp) * 10;
    ch_kp[8] = '0' + temp;

    temp = Ti;
    ch_Ti[5] = '0' + temp;
    ch_Ti[6] = '.';
    temp = (temp - (int)temp) * 10;
    ch_Ti[7] = '0' + temp;
    temp = (temp - (int)temp) * 10;
    ch_Ti[8] = '0' + temp;

    temp = Td;
    ch_Td[5] = '0' + temp;
    ch_Td[6] = '.';
    temp = (temp - (int)temp) * 10;
    ch_Td[7] = '0' + temp;
    temp = (temp - (int)temp) * 10;
    ch_Td[8] = '0' + temp;

    temp = yZad;
    ch_yZad[7] = '0' + (int)temp / 100;
    temp = temp - ((int)temp / 100) * 100;
    ch_yZad[8] = '0' + (int)temp / 10;
    temp = temp - ((int)temp / 10) * 10;
    ch_yZad[9] = '0' + (int)temp;

    TFTDisplay_ILI9341_String(210, 5, ch_kp);
    TFTDisplay_ILI9341_String(210, 20, ch_Ti);
    TFTDisplay_ILI9341_String(210, 35, ch_Td);
    TFTDisplay_ILI9341_String(210, 50, ch_yZad);
    break;

case 6:
    FunctionRealTime(t_a[0], t_H[0], t_x0[0], t_y0[0], v_u, H_MAX,
0);
    FunctionRealTime(t_a[0], t_H[0], t_x0[1], t_y0[0], v_e, H_MAX,
0);
    FunctionRealTime(t_a[0], t_H[0], t_x0[0], t_y0[1], v_yWy,
H_MAX, 0);
    WaterTank_Pipes(t_a[0], t_H[0], t_x0[1], t_y0[1], yWy,
yWyPrev, H_MAX, z, 0);
    Cross(0);

    TFTDisplay_ILI9341_String(165, 5, ch_e);
    TFTDisplay_ILI9341_String(5, 125, ch_y);
    TFTDisplay_ILI9341_String(5, 5, ch_u);

    break;
}
currentScreenNumber = screenNumber;
TimPrev = Tim;
}
}
}

```


StartTaskLED1

```
void StartTaskLED1(void *argument)
{
    /* USER CODE BEGIN StartTaskLED1 */
    /* Infinite loop */
    for (;;)
    {
        TFTDisplay_ILI9341_FillScreen(TFT_COLOR_ILI9341_RED);

        switch (screenNumber)
        {
            case 1:
                kp = t_kp[1];
                Ti = t_Ti[1];
                Td = t_Td[1];
                yZad = t_yZad[2];

                break;

            case 3:
                kp = t_kp[0];
                Ti = t_Ti[0];
                Td = t_Td[0];
                yZad = t_yZad[1];
                break;

            case 5:
                kp = t_kp[2];
                Ti = t_Ti[2];
                Td = t_Td[2];
                yZad = t_yZad[4];
                break;
        }
        ki = kp * Tp / Ti;
        kd = kp * Td / Tp;
        e = yZad - yWy;
        u = PID_Calculate(e, 0);
        yWyPrev = yWy;
        yWy = InercModel_Calculate(u) - z;
        if (yWy < 0)
        {
            yWy = 0;
        }
        if (iv == nv - 1)
        {
            for (int i = 0; i < nv - 1; i++)
            {
                v_u[i] = v_u[i + 1];
                v_e[i] = v_e[i + 1];
                v_yWy[i] = v_yWy[i + 1];
            }
            v_u[iv] = u;
            v_e[iv] = e;
            v_yWy[iv] = yWy;
        }
        else
        {
            v_u[iv] = u;
            v_e[iv] = e;
            v_yWy[iv] = yWy;
            iv++;
        }
        osDelay(Tp_ms);
        Tim++;
    }
    /* USER CODE END StartTaskLED1 */
}
```