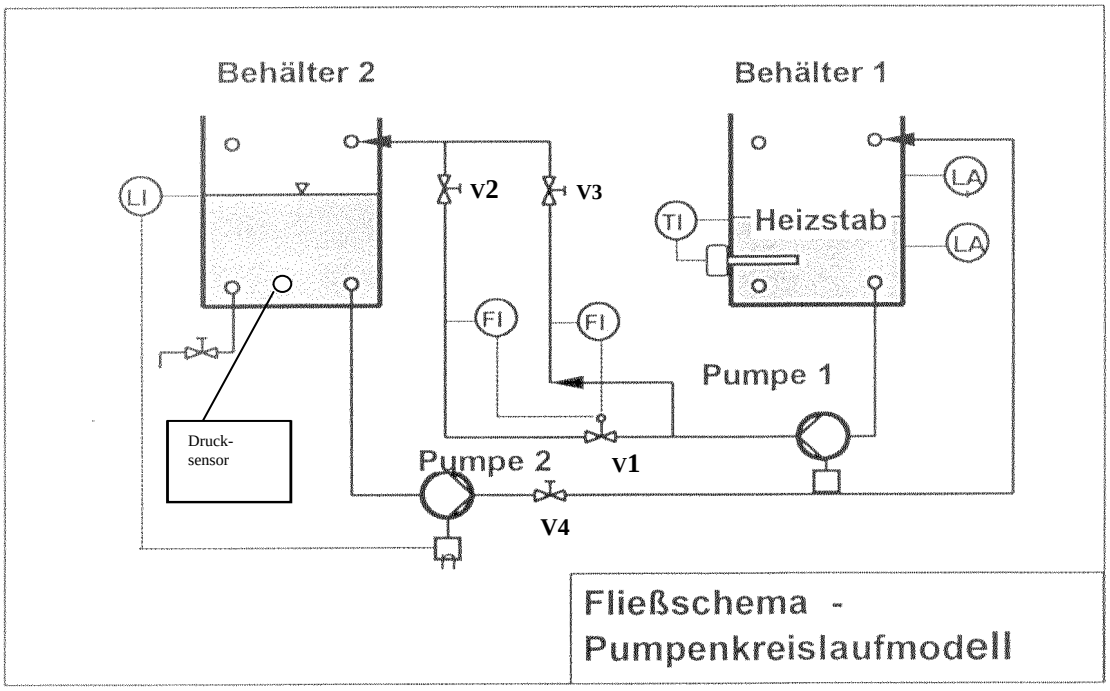
Programmcode:

```
/*-----------------------------------------------------------------------------*/
/* Programmbeschreibung - PID-Regelalgorithmus (Stellungsalgorithmus)          */
/*-----------------------------------------------------------------------------*/
/*Als Regelungsübung soll im Behälter 2 des Prüfstandes ein vorgegebenes       */
/*Niveau möglichst genau gehalten werden können (beispielsweise H=150 mm).     */
/*Von diesem Behälter 2 fördert eine Pumpe                                     */
/*P2 im Konstantbetrieb Wasser vom Behälter 2 in den Behälter 1. Umgekehrt     */
/*ist eine regelbare Pumpe P1 vorhanden, die vom Behälter 1 in den             */
/*Behälter 2 fördert. Diese Pumpe P1 soll über einen mA - Ausgang (0-20 mA)    */
/*als Stellglied von der SPS aus angesprochen werden können. Die Regelgröße    */
/*Niveau kann mittels Drucktransmitter (0 - 0,1 bar entsprechend 4-20 mA       */
/*am Ausgang) direkt in die SPS eingespeist werden.                            */
```




```
/*-----------------------------------------------------------------------------*/
/* DEFINE                                                                      */
/*-----------------------------------------------------------------------------*/
#define KONST 0.1/16*100000*1000/(1000*9.81)  /* Umrechnung Druck in Niveau */
#define SOLLNIVEAU_W 150.    /* Niveausollwert in mm Höhe eingeben !          */
#define YH 20.               /* Stellbereich in mA                            */
#define XH 200.              /* Messbereich d. Regelgröße in mm               */
#define RICHT 1.             /* -1= steigende Kennlinie,1=fallende Kennlinie */
                             /* (wenn de<0 soll auch dy<0 sein)               */
#define KPR 3.               /* Verstärkungsfaktor PID dimensionslos          */
#define TN  0.5              /* Nachstellzeit PID                             */
#define TV  0.1              /* Vorhaltezeit PID                              */
#define DT 0.1               /* Zeitschrittweite in sec eingeben !            */
                             /* Muß größer-gleich gewählter Zykluszeit sein !*/
/*-----------------------------------------------------------------------------*/

/*-----------------------------------------------------------------------------*/
/* VARIABLENDEKLARATIONEN                                                      */
/*-----------------------------------------------------------------------------*/
_GLOBAL INT pu1_io;
_GLOBAL INT druck_io;

_LOCAL TONtyp ton_dt;
```

```
_LOCAL REAL istniveau_x;
_LOCAL REAL e;       _LOCAL REAL e1;
_LOCAL REAL y;       _LOCAL REAL summe_e;
/*---------------------------------------------------------------------------*/
/* INIT - PROGRAMM                                                           */
/*---------------------------------------------------------------------------*/
_INIT void initteil (void)
{
        ton_dt.PT=DT*1000.;         /* Zeitschrittweite - Zeitzuweisung     */
                                    /* Faktor*1000, da intern in ms         */
        ton_dt.IN=1;                /* Start erster Zyklus                  */

        istniveau_x=0.;
        e=e1=0.;
        y=0.;
        summe_e=0.;
}
/*---------------------------------------------------------------------------*/
/* ZYKLISCHES PROGRAMM                                                       */
/*---------------------------------------------------------------------------*/

_CYCLIC void zyklischteil (void)


{
TON(&ton_dt);                               /* Start Zeitschritt            */

    if (ton_dt.Q==1)                        /* Zeitschritt abgelaufen       */
    {
        ton_dt.Q=0;                         /* Rücksetzen Zeitbaustein.Q    */
        ton_dt.M=0;             /* damit sofortiger Neustart möglich ist    */


        istniveau_x=((REAL)druck_io*20./32767.-4)*KONST;    /*Istniveau     */

        e=SOLLNIVEAU_W-istniveau_x;                         /*Regeldifferenz*/

        summe_e= summe_e+e;

        y=RICHT*YH/XH*KPR*(e+ (DT*summe_e/TN) + (TV/DT * (e-e1)));


        if (y<0)y=0;            /* Stellgrenze Minimum  */
        if (y>YH) y=YH;         /* Stellgrenze Maximum  */

        e1=e;

        pu1_io= (INT)(y*32767/20.); /* Skalierung I/O-Modul anpassen        */

        ton_dt.IN=1;                /* Neustart Zeitbaustein                */
    }
}
```