

```

1 // UserConfiguration.h
2 //
3 //
4 /*=====
5 /*=====
6 /*=====  START USER CONFIGURATION FOR THE I/O  =====
7 /*=====
8 /*=====
9 //
10 // Define all used pins for the application.
11 //
12 // The range of digitals is commonly used. First to define the inputs,outputs and ultrasonics.
13 // The analog out en servo pins are in the PWN range of the board.
14 //
15 const byte NoInputPins    = 5;
16 const byte FirstInput     = 22;
17 const byte NoOutputPins   = 10;
18 const byte FirstOutput    = 30;
19 const byte NoUltrasonic   = 1;
20 const byte FirstUltra     = 46;
21 const byte NoAnaInPins    = 0;
22 const byte FirstAnaIn     = 0;
23 const byte NoAnaOutPins   = 0;
24 const byte FirstAnaOut    = 0;
25 const byte NoServos       = 0;
26 const byte FirstServo     = 0;
27 const byte NoTemp         = 0;
28 const byte OneWireChannel = 0;
29 const byte NoMarkers      = 1;
30 //
31 // IMPORTANT!
32 //
33 // No. of tagnames have to be equal with No. of pins! This is the addressing part between the pins and the tags!
34 // Tagnames are connected with the in/outputs by their order. When done correctly it is possible to programm all
35 // software by tags and phases in an easy way. Don't remove Tag lines (if not used keep them empty)!

```

```
36 //
37 String DummyTags[] = { "" };
38 String InputTags[] = { "F1", "F2", "F3", "START", "ZERO" };
39 String OutputTags[] = { "&D1", "&D2", "&D3", "&DW", "ACKD1", "ACKD2", "ACKD3", "ACKSTART", "READY", "&CIP" };
40 String AnaInTags[] = { "" };
41 String AnaOutTags[] = { "" };
42 String ServoTags[] = { "" };
43 String UltraTags[] = { "PRESENT" };
44 String MarkerTags[] = { "MUZIEK" };
45 String TempTags[] = { "" };
46 //
47 // Declaration of all used states in the FSM.
48 // The first and last state "START" and "END" are obligatory and may not be removed, the rest is up to you
49 //
50 String PossibleFSMStates[] = { "START",
51                                "INIT", "CHOICE", "GOTO-LIQUIR", "LIQUIR", "GOTO-WATER", "WATER", "READY",
52                                "MIX", "NEWCUP", "CUPPRESSENT", "NORMAL", "SERVICE", "CIP", "CIPCYCLE",
53                                "END" };
54 //
55 // Start defining User timers don't use predefined timers (stay out of range 200-300)
56 //
57 const byte timAanUit = 1;
58 const byte timWait = 2;
59 const byte timLimo = 3;
60 const byte timWater = 4;
61 const byte timTask4 = 5;
62 const byte timStartup = 6;
63 const byte timGameOver = 7;
64 const byte timService = 8;
65 const byte timServiceMode = 9;
66 const byte timWaitCIP = 10;
67 const byte timCIP = 11;
68 //
69 // End defining user timers
70 //
```

```

71 // START USER SPECIFIC DECLARATIONS
72 //
73 // Start User variables:
74 //
75 boolean AanUit;
76 int LiquirNo;
77 int PositionLiquir;
78 int TimeLiquir;
79 boolean NewCup;
80 boolean CupPresent;
81 boolean GoMix;
82 boolean PrepareDrink = false;
83 boolean Service = false;
84 int TimeNoCupPresent;
85 int PresentCount;
86 int ServiceLED = 1;
87 //
88 // End User variables:
89 //
90 // use of the extra MEGA board for the messages, use MessageI2C and/or UseHMISerial, do not remove, only set true or false
91 //
92 boolean UseI2C = false;
93 boolean UseHMISerial = true;
94 boolean UseGPS = false;
95 boolean UseIntercard = false;
96 boolean PollUltra = false;
97 //
98 /*=====
99 /*=====
100 /*=====  END USER CONFIGURATION =====
101 /*=====
102 /*=====

```