Expertise
and insight
for the future

Arsi Arola, Duc Vo, Mikko Larke

Ventilation project

Metropolia University of Applied Sciences

Smart systems

Project documentation

13 March 2020

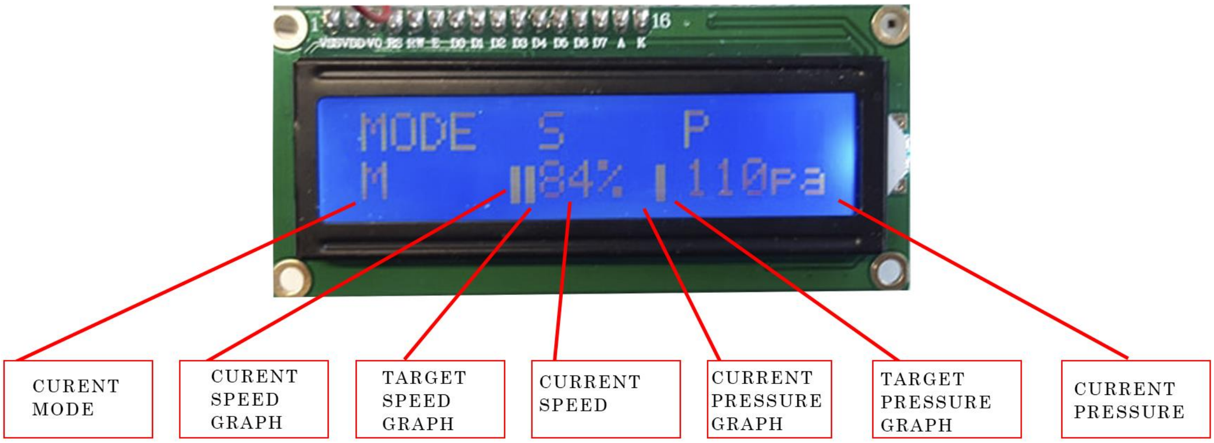
Authors Title Number of Pages Date	Arsi Arola, Duc Vo, Mikko Larke Ventilation project 11 pages 13 March 2020
Degree	Bachelor of Engineering
Degree Programme	Information technologies
Professional Major	Smart systems
Instructors	Keijo Länsikunnas

Simple ventilation system where user can set fan speed manually or use pressure readings to set fan speed automatically to match pressure reading inside vent.

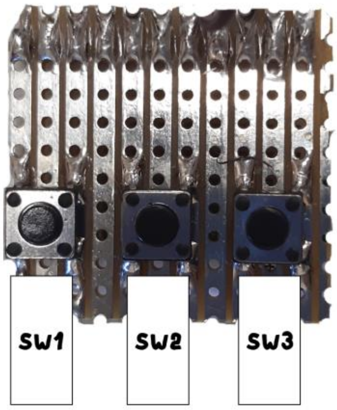
Contents

1 User manual

1.1 UI overview

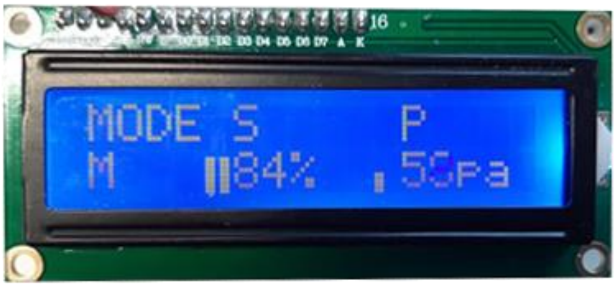


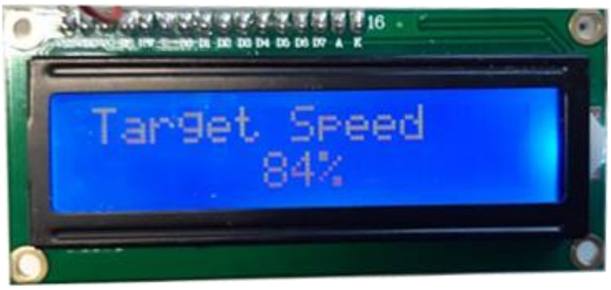
1.2 Button functionality



Button	Manual mode	Automatic mode	Error message
SW1	Lower target speed	Lower target pressure	Clear error message
SW2	Change to automatic mode	Change to manual mode	Clear error message
SW3	Increase target speed	Increase target pressure	Clear error message

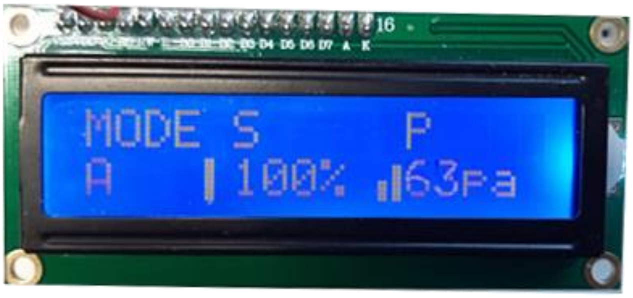
1.3 Manual mod

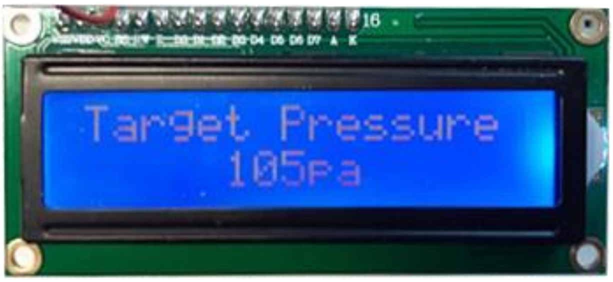




In manual mode user can set target fan speed and it is indicated by letter M in displays lower left corner. While in manual mode home screen shows two bars in front of current fan speed. Left bar indicates actual fan speed and right bar target fan speed. Fan speed is measured in percent’s in between 0 (fan doesn’t spin) and 100 (fan spins at maximum speed). Speed is set by buttons SW1 (lowers target speed) and SW3 (increases target speed). Pressing SW2 confirms change and sets fan speed.

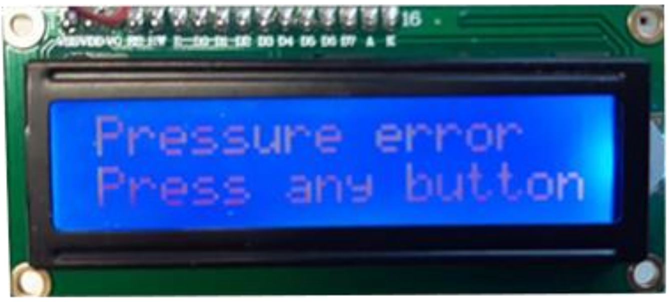
1.4 Automatic mode





In automatic mode user can set target pressure and fan speed will be automatically adjust to actual pressure. Automatic mode is indicated by letter A in displays lower left corner. While in automatic mode home screen shows two bars in front of current pressure. Left bar indicates actual pressure and right bar pressure. Pressure is measured in pascals in between 0 (lowest pressure reading) and 120 (highest pressure reading). Pressure is set by buttons SW1 (lowers target pressure) and SW3 (increases target pressure). Pressing SW2 confirms change and sets pressure.

1.5 Pressure error message



```
Pressure error
Press any button
```

In automatic mode if target pressure can’t be reached within 10 seconds will display show error message. Press any button to clear message.

2 Wiring

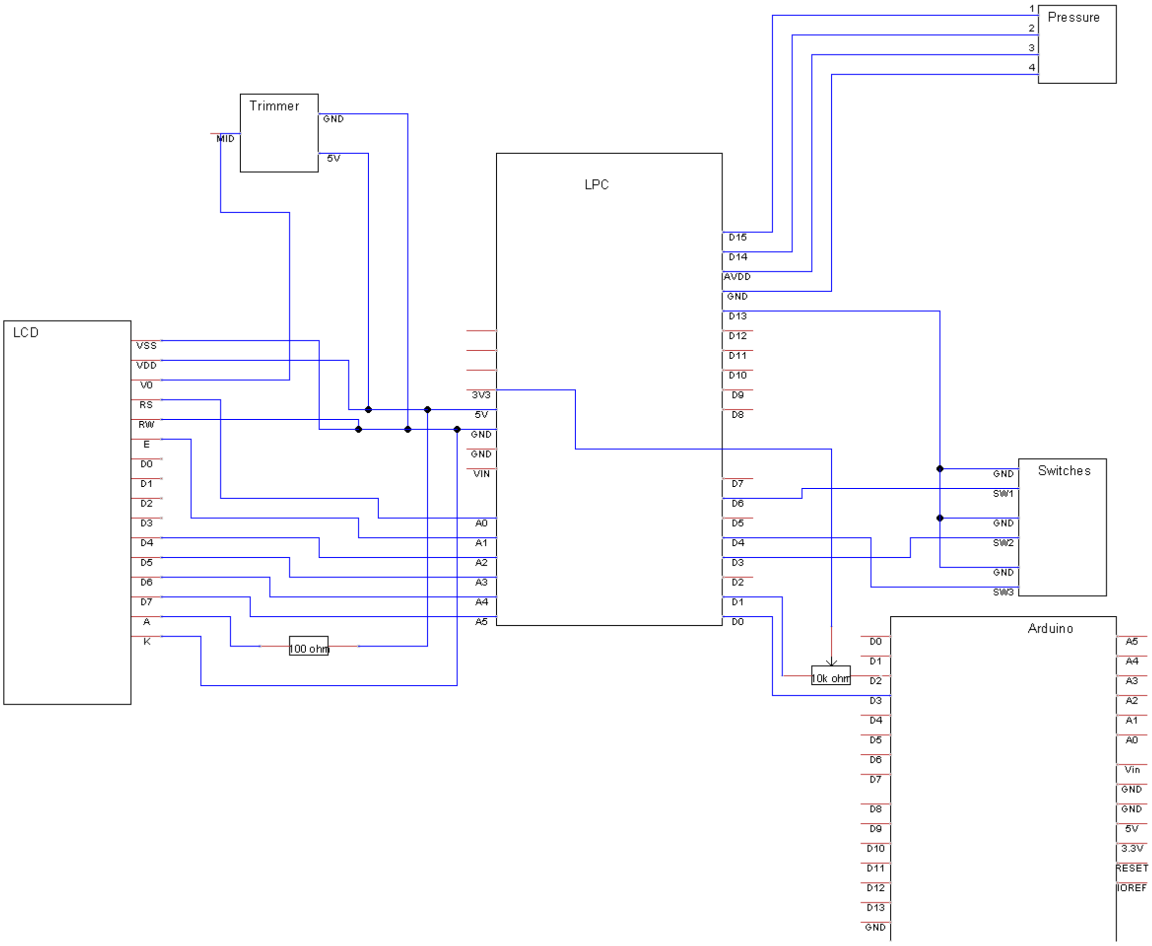
2.1 Wiring table

Table of used peripherals and on which Lpc pin they are connected. Red rows indicate free pins on Lpc.

Lpc	LCD	Buttons	Pressure	Modbus	Trimmer	Lpc return
N/A						
IOREF						
NRST						
+3V3				Digital 2 Resistor		D1
+5V	VDD, A(+resistor)				Lower pin	
GND	R/!W, VSS, K	SW1-GND SW2-GND SW3-GND		Arduino Ground	Upper pin	
GND						
VIN						
A0 (0_8)	RS					
A1 (0_7)	E					
A2 (0_6)	DB4					
A3 (0_5)	DB5					
A4 (0_23)	DB6					
A5 (0_22)	DB7					
D0				Digital 3		
D1				Digital 2 (Same as 3.3v)		
D2						

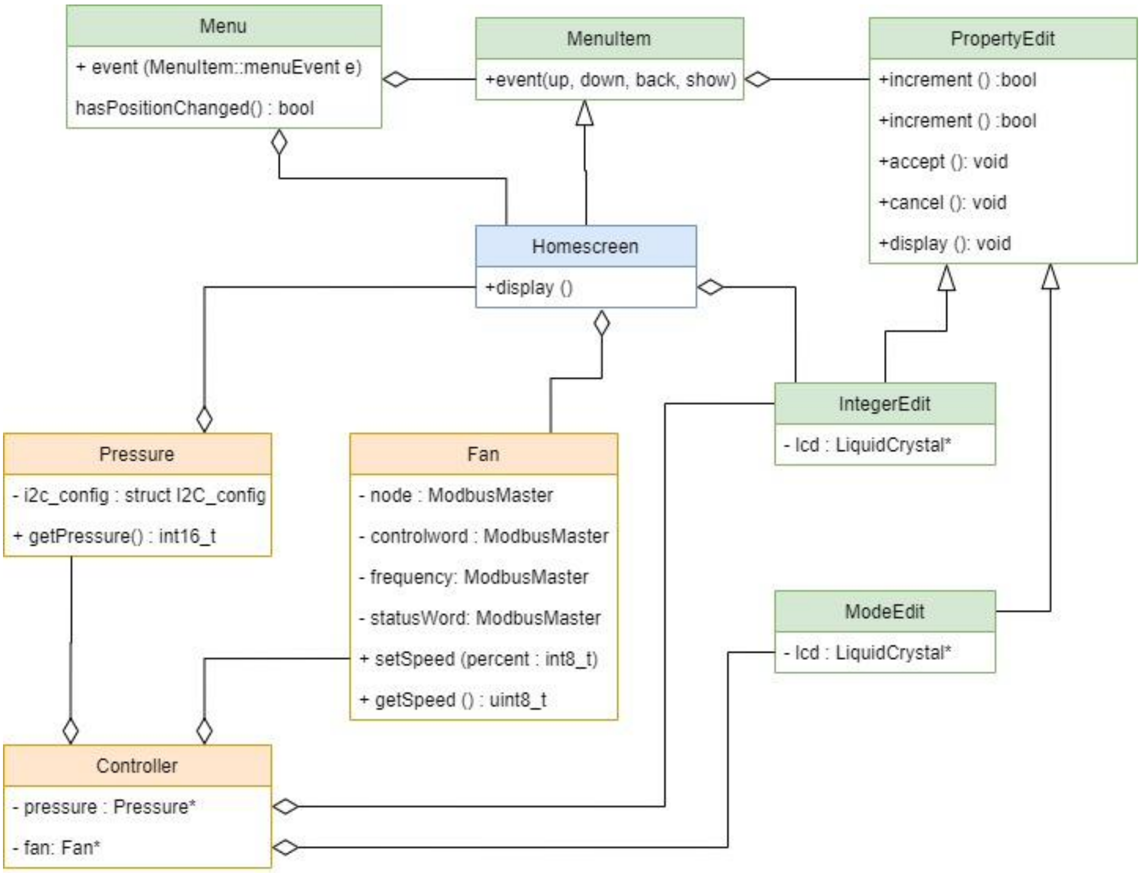
D3	Sw2					
D4	Sw3					
Lpc	LCD	Buttons	Pressure	Modbus	Trimmer	Lpc return
D5						
D6	Sw1					
D7						
D8						
D9						
D10						
D11						
D12						
D13						
GND			Pressure			
AVDD			Pressure			
D14			Pressure			
D15			Pressure			
LCD trimmer						
Center pin	V0					

2.2 Wiring diagram



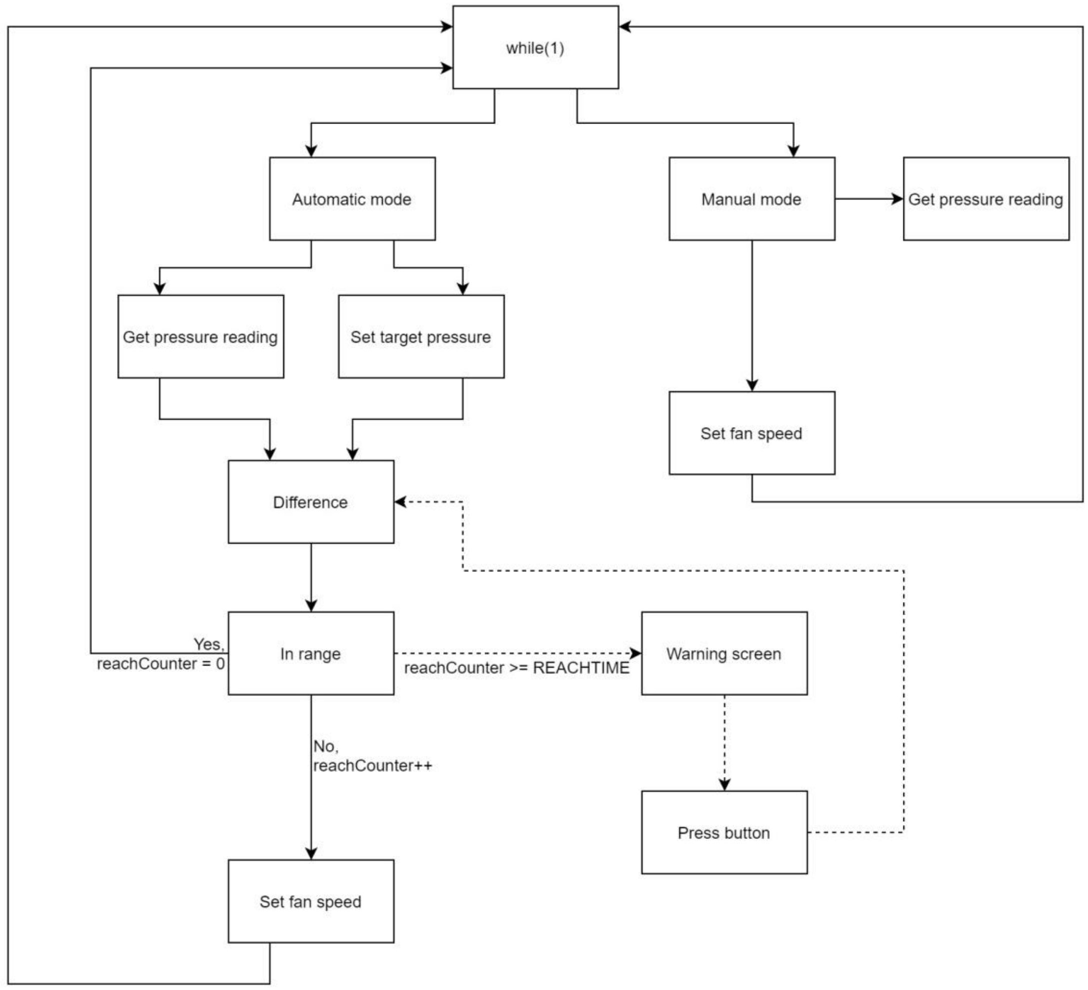
3 Project documentation

3.1 Simplified class diagram



Full class diagram can be found at https://github.com/arsiarola/ventilation-project/blob/master/img/ventilation-uml.png

3.2 Flowchart



3.3 Calculating fan speed in automatic mode

In automatic mode pressure inside vent should stay at user given value, which is implemented with spinning the fan at certain speed. Increasing fan speed increases pressure inside the vent. Because of that we can calculate the difference between user given target pressure and sensors pressure reading to adjust the fan speed. If the value is positive, we increase current fan speed with value and with negative values we subtract current fan speed with value.

$$difference = TargetPressure - ActualPressure$$

Testing this formula gets to correct pressure over time, but oscillation around correct value was too large. To decrease the oscillation, we used square root of the absolute difference and declared the sign of difference according to subtraction.

$$\begin{Bmatrix} difference = -\sqrt{|difference|} & , difference < 0 \\ difference = \sqrt{|difference|} & , difference \geq 0 \end{Bmatrix}$$

With this the oscillation around correct value is fine, but even small pressure differences adjusted the fan and caused pressure to change from correct value too easily. To prevent that we implemented isInRange function to check if pressure difference is within set value and use that to prevent changing fanspeed when true.

```
bool isInRange(int range)
{
    return (abs(pressureDifference()) < range);
}
```