Thanh Huynh – 2688093

Bilal Butt – 2688700

Michel Clerger – 2694646

Zirui Song - 2652528

Design document

GROUP MEMBERS:

Contents

[CLASS DIAGRAM 2](#_Toc468452010)

[Class diagram 2](#_Toc468452011)

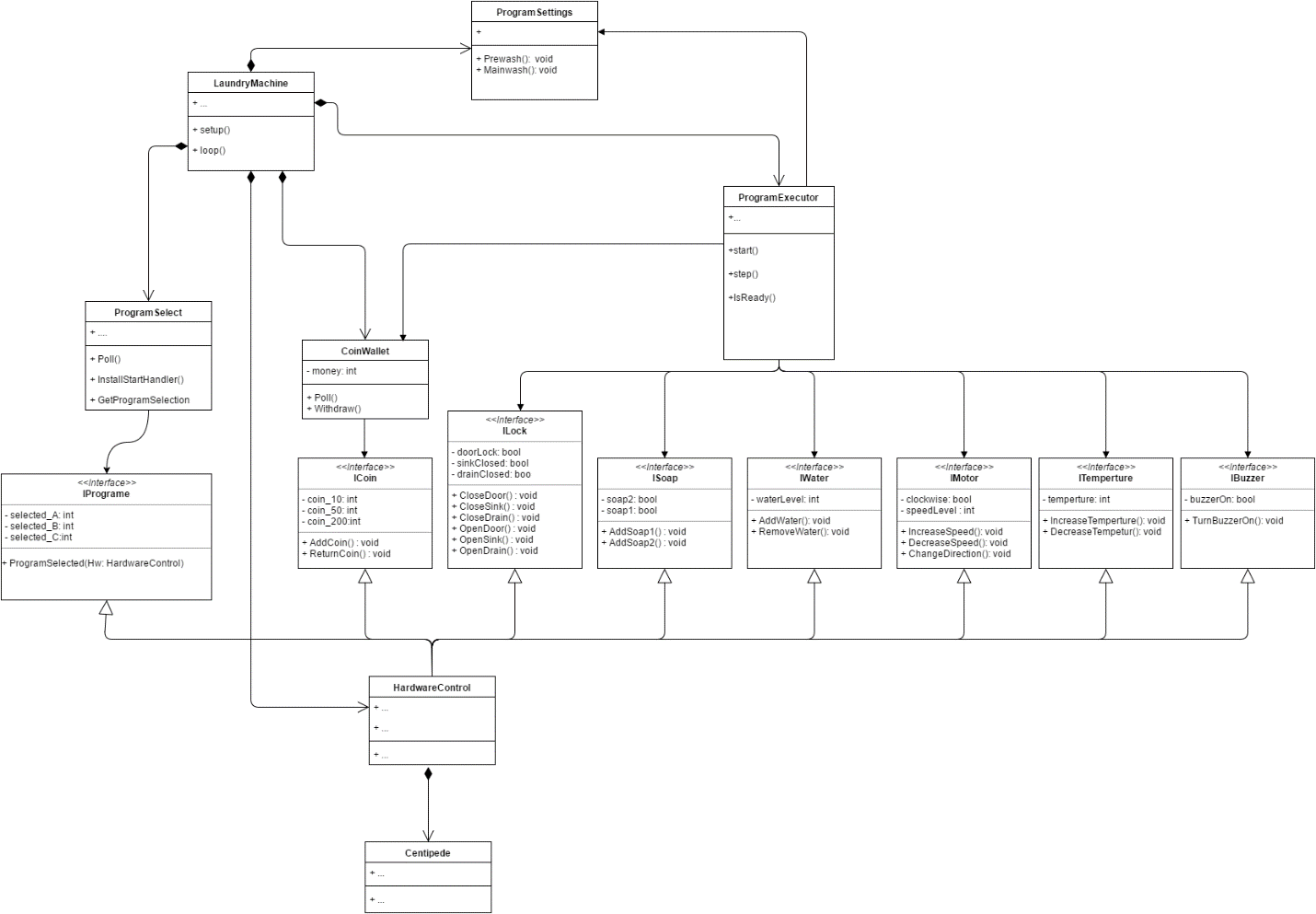
[Description of the classes and their members 2](#_Toc468452012)

[STATE DIAGRAMs 3](#_Toc468452013)

[PIN CONFIGURATIONS 3](#_Toc468452014)

# CLASS DIAGRAM

## Class diagram



## Description of the classes and their members

By default, all of the properties of classes are private and methods of them are public. If it is not the case, the visibilities of these properties and methods will be indicated.

By default, all of the member variables of classes are private and their methods are public. If it is not the case, the visibilities of these properties and methods will be indicated. The class diagram is still subject to refinement and will be amended as deemed necessary.

The classes as depicted in the class diagram are based on the notion of how the Arduino itself operates. Since the Arduino does not have an operating system instead it operates by consistently running a main loop that continues indefinitely. Therefore in the class hierarchy the LaundryMachine class plays the role of the class that contains the main loop.

The loop of the LaundryMachine class is constantly polling the ProgramSelect and CoinWallet classes to see if they have received any input from the user indicated by changes in their respective member variables. Once it has been determined if the program selected and the relevant payment has been completed and the user has indicated that he wants to begin the washing program, the ProgramExecutor will be called.

The ProgramExecutor executes the complete washing program for the washing program setting initially identified by the user. The interface classes are implemented in the HardwareControl class and are responsible for implementing functions that have to do with manipulating the actual hardware.

## HARDWARE RELATED CLASSES

#### ICOIN

ICoin is the interface class containing the virtual functions relating which is implemented in the hardware class.

|  |  |
| --- | --- |
| ICOIN | |
| Functions | Description |
| *virtual boolean GetCoin10Button() = 0;* | This is used for…. |
|  |  |
|  |  |
| Remarks:  (Please describe here any problems, remarks, current states of your functions) | |

# STATE DIAGRAMs

## PIN CONFIGURATIONS

From our independent investigations we tried to determine which buttons correspond to which exact input descriptors. In the hardware control class the constants used for each of the corresponding buttons and switches were as follows:

Keyselect must be HIGH in order to read button inputs.

Button 1: IN\_IN0 and IN\_IN3, pressing button 1 raises IN0 and IN3 to high  
Button 2: IN\_IN0, pressing button 2 raises IN0 to high  
Button 3: IN\_IN3, pressing button 3 raises IN3 to high  
Button 4: IN\_IN2, pressing button 3 raises IN2 to high  
Button 5: IN\_IN1, pressing button 3 raises IN1 to high  
Button 6: IN\_IN1, IN\_IN2 and IN\_IN3, pressing button 1 raises IN1, IN2 and IN3 to high

Keyselect must be LOW in order to read switch inputs.

Switch 2: IN\_IN0, pulling up switch 2 raises IN0 to high  
Switch 3: IN\_IN3, pulling up switch 3 raises IN3 to high  
Switch 4: IN\_IN1, pulling up switch 4 raises IN1 to high  
Switch 5: IN\_IN2, pulling up switch 5 raises IN2 to high

*Figure 1: Buttons and Switches on Laundry Machine*