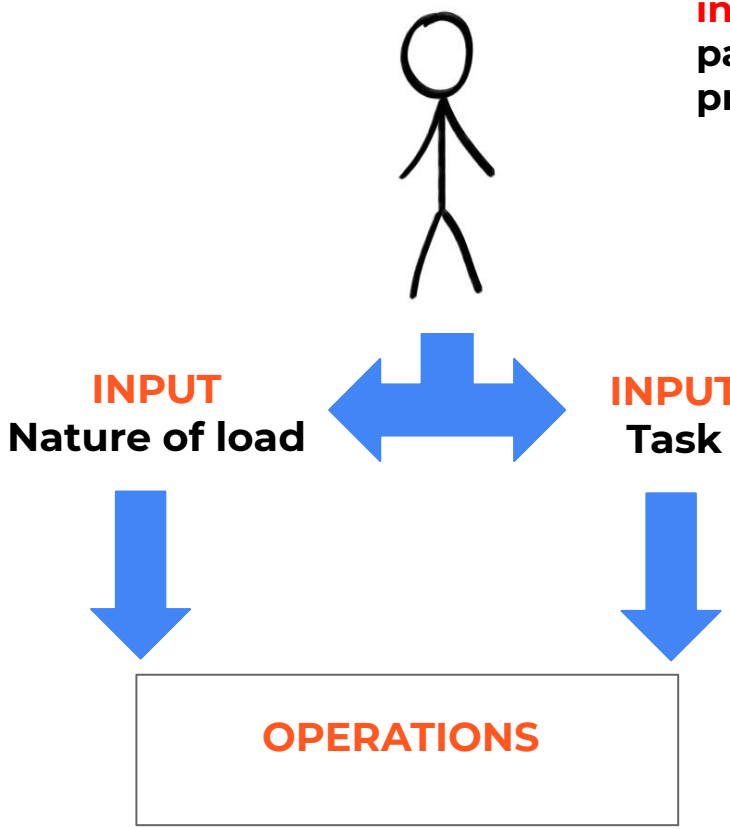
WASHING MACHINE

Functioning

Normal execution **can be interrupted** in case a pause button is pressed/lid is open.



START/PAUSE button

Operations

Water is filled at time of start

↓

00:15:00 Clothes are soaked in detergent water

↓

00:15:00 First wash

↓

00:04:00 Water is drained for the first time

↓

00:10:00 First rinse (=second wash)

↓

00:04:00 Water is drained for the second time

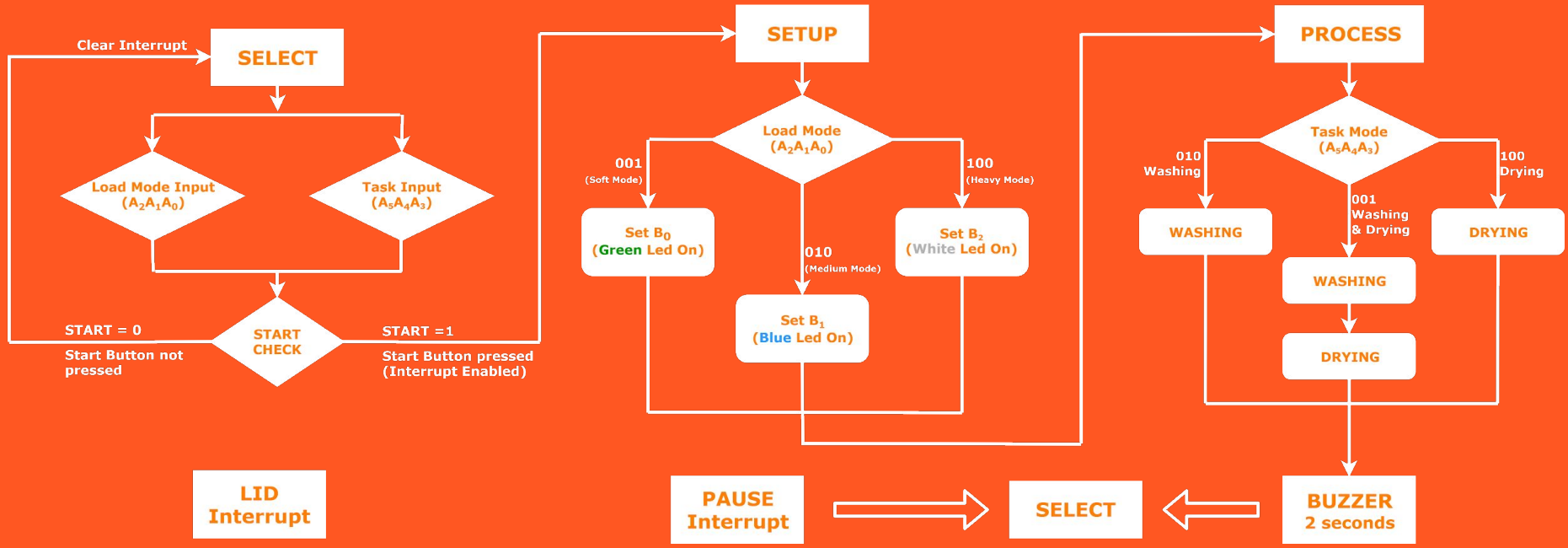
↓

00:05:00 Spin cycle

↓

00:00:15 BUZZER

Flowchart


Clear Interrupt
SELECT
Load Mode Input
($A_2A_1A_0$)
Task Input
($A_5A_4A_3$)
START CHECK
START = 0
Start Button not pressed
START =1
Start Button pressed
(Interrupt Enabled)
SETUP
Load Mode
($A_2A_1A_0$)
001
(Soft Mode)
010
(Medium Mode)
100
(Heavy Mode)
Set B_0
(Green Led On)
Set B_1
(Blue Led On)
Set B_2
(White Led On)
PROCESS
Task Mode
($A_5A_4A_3$)
010
Washing
001
Washing
& Drying
100
Drying
WASHING
WASHING
DRYING
DRYING
BUZZER
2 seconds
SELECT
PAUSE
Interrupt
LID
Interrupt

Pseudo Code

List of Pins and Registers of Atmega8 used:

- **PD2 port: [Acts as Start button when I=0] / [Works as Pause Interrupt (INT0) when I=1]**
- **PD3 port: Lid Open Interrupt (INT1)**
- **PORTB (the first 6 least significant bits) gives outputs that last during the working of the washing machine**
- **PORTC (the first 3, 5th and 6th significant bits) gives outputs indicating the load or task mode chosen; only when the button is pressed.**
- **PINA (the first 6 least significant bits) contains the inputs from the users.**
- **R16 (for AVR Assembly code) / Reset (int variable for C interfacing): Acts as a flag; set to 1 when the PAUSE ISR is called.**

When Global Interrupt is set:

- **Enters PAUSE (ISR) when INT0 goes from low to high level**
- **Enters LID (ISR) when INT1 is in the low level**

SELECT:

- **Global Interrupt disabled.**
- **PORTC=PINA; to display to the user which mode he chooses.**
- **Get inputs $A_2A_1A_0$ and $A_5A_4A_3$ from the user through ports PA2,PA1,PA0 and PA5,PA4,PA3, which would get correspondingly get reflected in PINA.**
- **Check PD2 (START). If PD2=1, go to SETUP. Else go to SELECT.**

Pseudo Code

SETUP:

- Global Interrupt is set.
- If the 3 least significant bits of PINA is
  a) 001 (Soft Mode): PORTB=0x01 (Green Led glows)
  b) 010 (Medium Mode): PORTB=0x02 (Blue Led glows)
  c) 100 (Heavy Mode): PORTB=0x04 (White Led glows)
- Check if R16 / Reset is 1, if yes, go to SELECT.
- Go to PROCESS

PROCESS:

- Check the 4-6th least significant bits. If the sequence is
  a) 010: go to WASHING
  b) 001: go to WASHING
     go to DRYING
  c) 100: go to DRYING
- PORTB=0x08 (Buzzer rings for 2s).
- Go to SELECT

Pseudo Code

WASHING:

- If the 3 least significant bits of PINA is
  a) 001 (Soft Mode): PORTB=0x31 (2 LEDs glow 3 times with a delay of 2s in addition to the Green LED)
  b) 010 (Medium Mode): PORTB=0x32 (2 LEDs glow 5 times with a delay of 2s in addition to the Blue LED)
  c) 100 (Heavy Mode): PORTB=0x34 (2 LEDs glow 7 times with a delay of 2s in addition to the White LED)
- In each loop for delay, R16 / Reset is checked. If it is 1, go to SELECT

DRYING:

- If the 3 least significant bits of PINA is
  a) 001 (Soft Mode): PORTB=0x31(2 LEDs glow for 4s in addition to the Green LED)
  b) 010 (Medium Mode): PORTB=0x32 (2 LEDs glow for 8s in addition to the Blue LED)
  c) 100 (Heavy Mode): PORTB=0x34 (2 LEDs glow for 12s in addition to the White LED)
- In each loop for delay, R16 / Reset is checked. If it is 1, go to SELECT

Pseudo Code

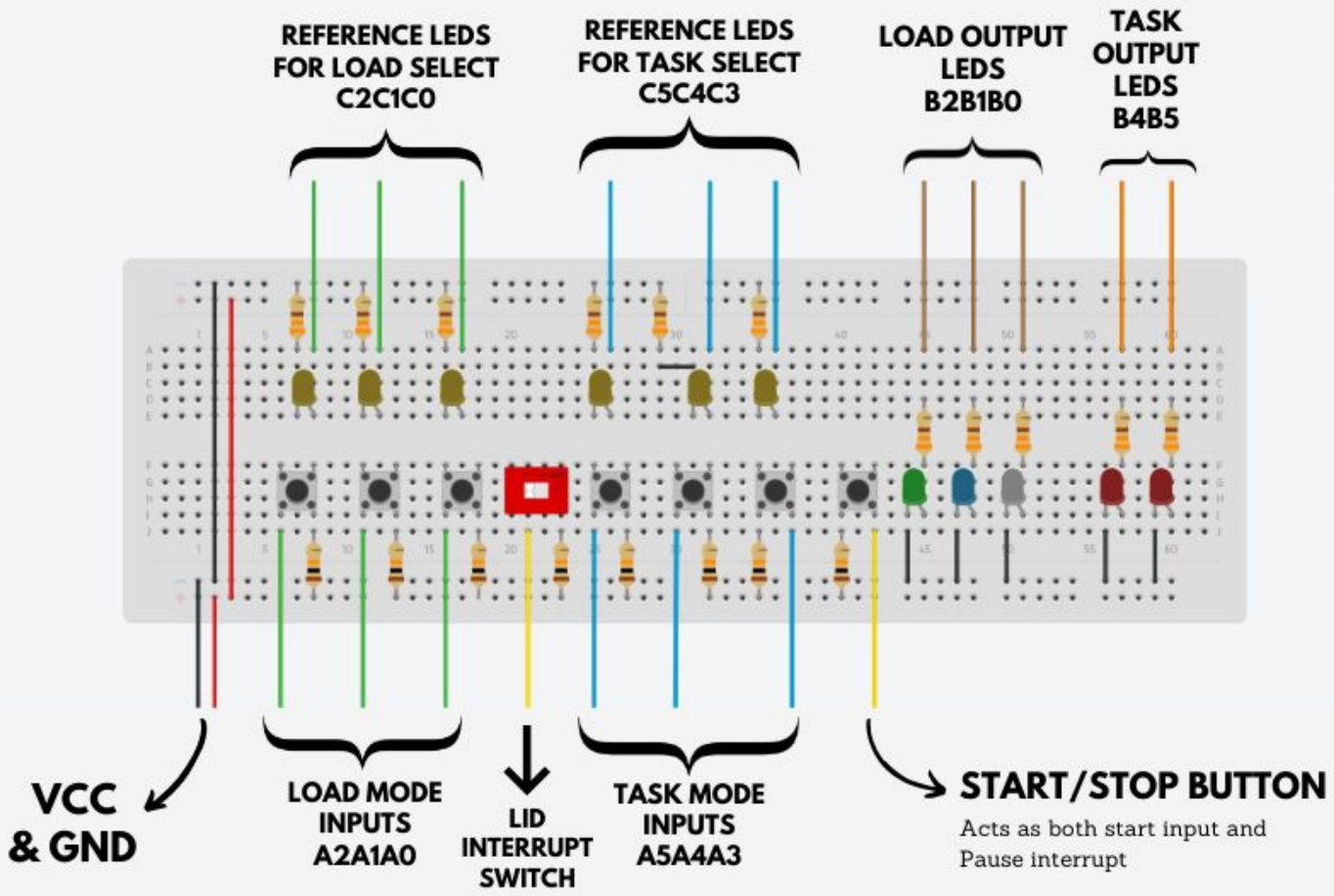
PAUSE (ISR):

- **Reset PINA to 0x00**
- **Set R16 / Reset=1.**
- **Returns back to where it left from. (From that part of the code, the PC goes to SELECT)**

LID (ISR):

- **Delay of 1s before checking whether the lid has been closed again.**
- **Returns when INT1=1**

Circuit Plan



Challenges

- **2 interrupts are used:** One for the Lid open detection and another for Pause button. But there is only a common interrupt enable bit. So, the same pin has to double up as an input pin(START) and an interrupt pin(PAUSE).

- The user has to give 2 sets of input: Task process and Load condition. The order in which the user gives the inputs shouldn't matter. The user should also be able to change his inputs before starting the machine. So we had to design a mechanism for taking parallel inputs using a flag bit (START). The user can change his inputs before pressing the start button, which on pressing, begins the operation.

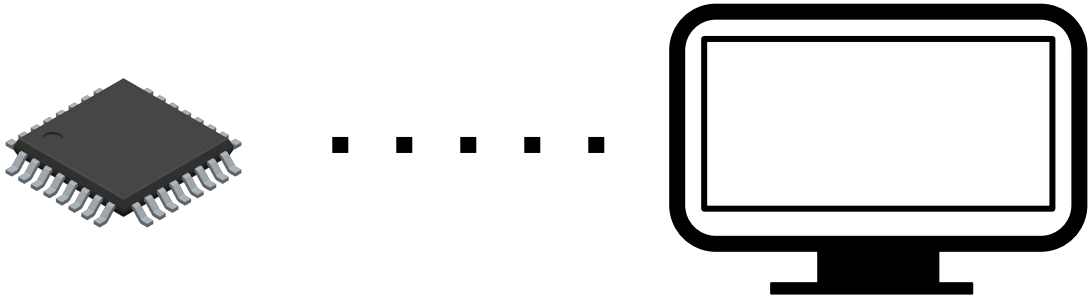
Challenges

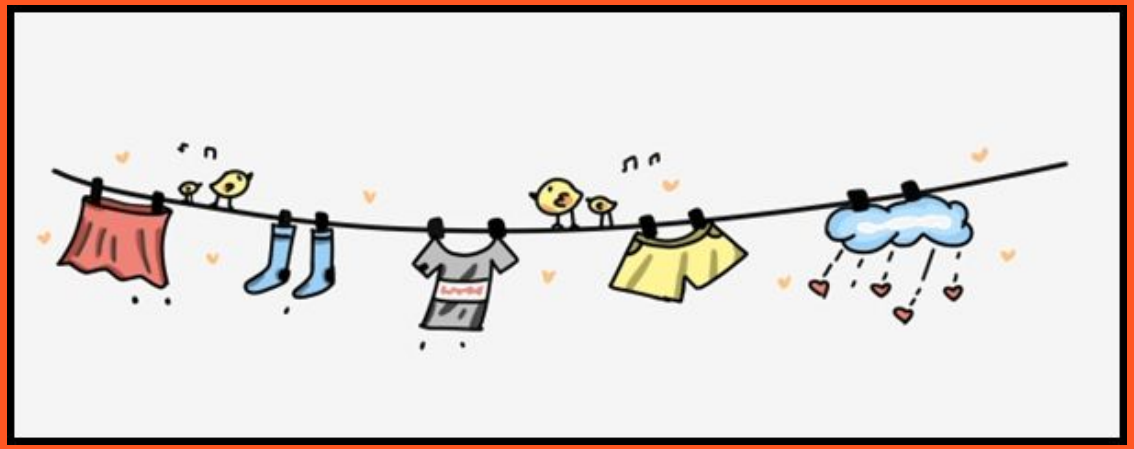
- The Pause interrupt should stop the ongoing program and reset the circuit to its initial values. But, the ISR cannot accommodate the jump to the beginning of the program as it will lead to an infinitely proceeding ISR. So, we had to update a flag bit (reset bit) in the ISR and poll for the flag bit in the main program.

- To simulate the interface of the actual washing machine, we have used push buttons for the inputs. If the user presses 2 different buttons simultaneously, for eg., SOFT and HEAVY modes, the last button to be pressed will be considered. It is assumed that both the buttons are not released simultaneously.

Possible Upgrades

- **Along with LEDs, we will try to show the process status on the desktop screen using serial communication in AVR.**
- **This can be further upgraded by displaying to an external LCD screen.**
- **We will try to demonstrate the physical working of a washing machine using a stepper motor and relays.**





EE18B002: **AbishekShivram A M**

EE18B006: **Gayatri Ramanathan Ratnam**

EE18B032: **Sidesh S**

EE18B038: **Srivenkat A**

EE18B120: **Sarvani Cheruvu**

EE18B132: **G K Hemanth Ram**