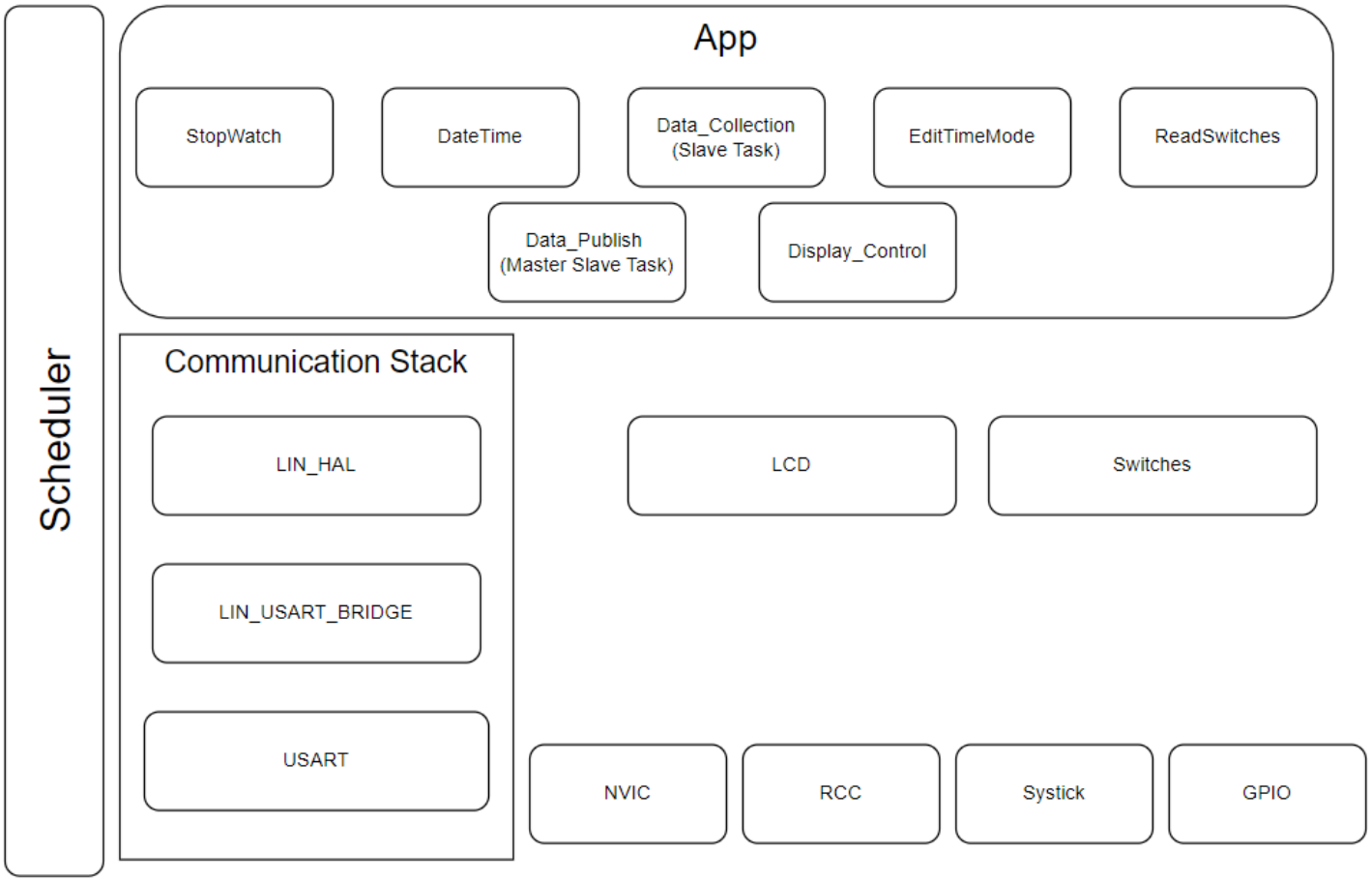
Demo1
Static Analysis

1-Components Diagram

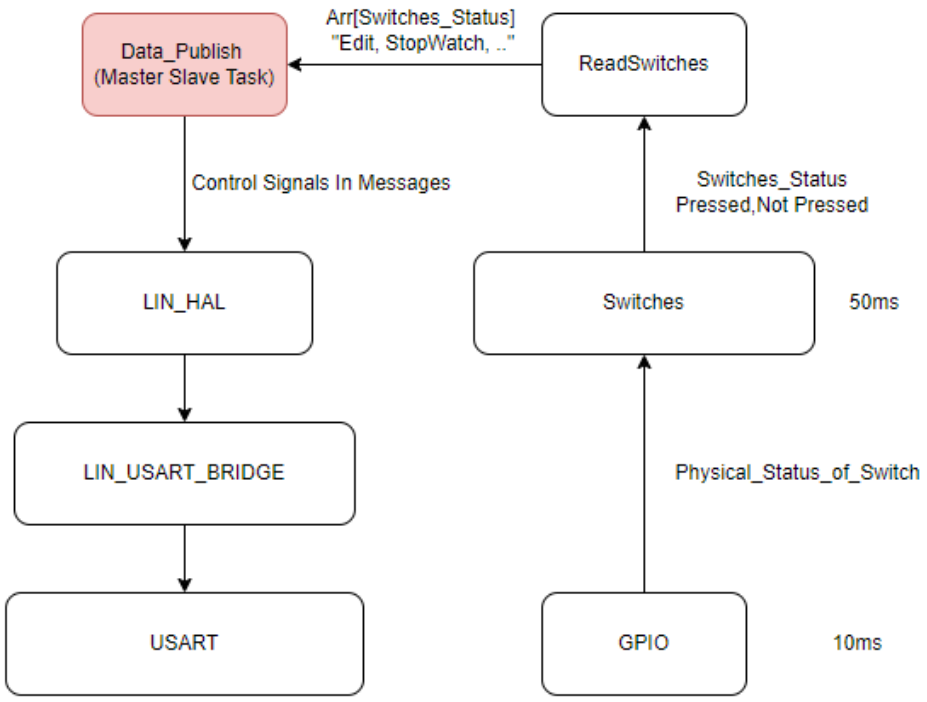


Our application comprises seven modules:

- Stopwatch: Controls the stopwatch mode.
- DateTime: Manages the date (02/04/2024) and current time (09:30:25) display.
- Data Collection (Slave Task): Collects data via the LIN protocol.
- EditTimeMode: Controls the Edit Mode.
- ReadSwitches: Retrieves switch values.
- Data Publish (Master Slave Task): Sends data over LIN protocol.
- Display Control: Selects the appropriate mode for LCD display.

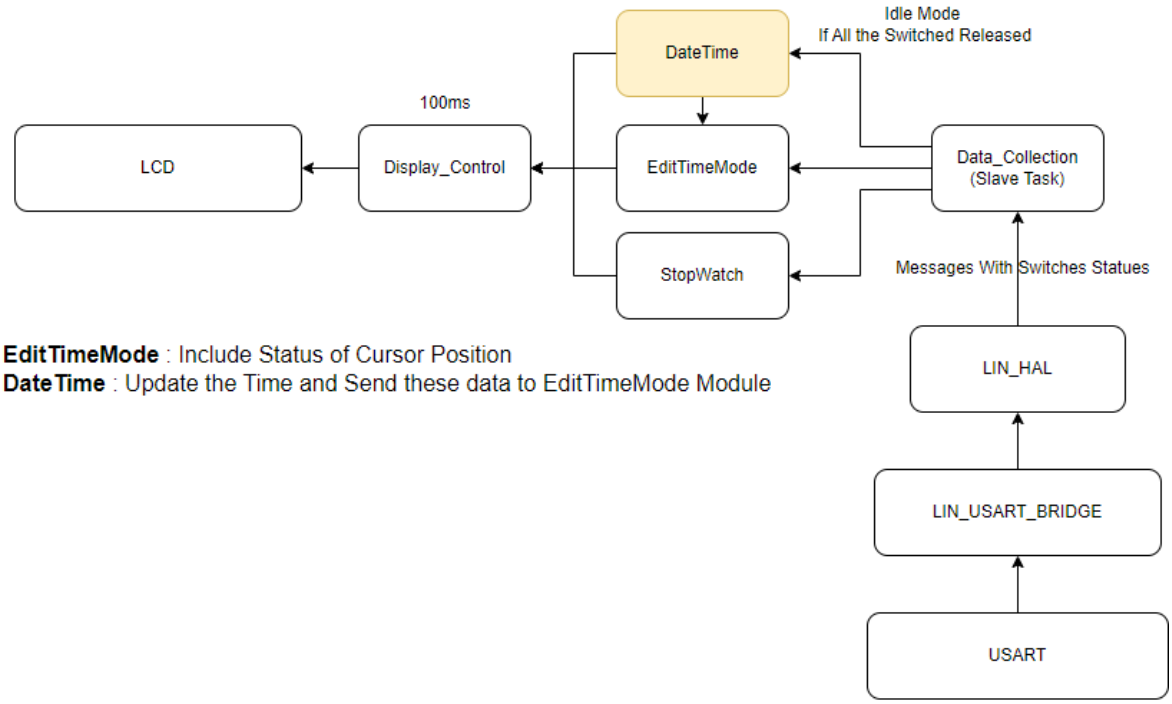
2-Data Flow Diagram

2.1 Master MC:



- ReadSwitches will send the readings of the switches from Master MC and send the data to Data_Publish

2.2 Slave MC:



EditTimeMode : Include Status of Cursor Position
DateTime : Update the Time and Send these data to EditTimeMode Module

- DateTime, EditTimeMode and StopWatch will run in parallel.
- DateTime and StopWatch Modules Running in background all the time.
- Data_Publish will collect the signals in one message to be sent to LIN (Master Slave Task).

3-Bus Matrix

3.1 Scheduler Table

Msg Name	**Time Slot**	**Msg ID**	**Master**	**N0**
Mstr_Ctrl	TBC	0x05	W	R
Node_Ctrl	TBC	0x07	R	W

3.2 Signals

3.2.1 Signals in Mstr_Ctrl

	Up_M	**Down_M**	**Left_M**	**Right_M**	**Ok_M**	**Mode_M**	**Edit_M**	**Start_M**	**Stop_M**
Start	0	1	2	3	4	5	6	7	8
Len	1	1	1	1	1	1	1	1	1
Msg	0x05	0x05	0x05	0x05	0x05	0x05	0x05	0x05	0x05

3.2.2 Signals in Node_Ctrl

	Up_N0	**Down_N0**	**Left_N0**	**Right_N0**	**Ok_N0**	**Mode_N0**	**Edit_N0**	**Start_N0**	**Stop_N0**
Start	0	1	2	3	4	5	6	7	8
Len	1	1	1	1	1	1	1	1	1
Msg	0x07	0x07	0x07	0x07	0x07	0x07	0x07	0x07	0x07

4. Tasks Distribution

Team1 : Moamen Hamed – Momen ElSayed

Team2: Ahmed Osman – Mohammed Ebrahim

Momen Elsayed – Ahmed Osman	Moamen Hamed – Mohammed Ebrahim
GPIO	Systick
RCC	NVIC
USART	Scheduler
LIN	LIN_UART_Bridge
Switches	LCD
EditTimeMode	ReadSwitches
Data_Publish	Data_Collection
StopWatch	DateTime

5. Notes

if (time >= 12 Apr.2024)

{

We will work with **USART protocol** and we will send data when the switch is pressed;

}

else

{

We will work with **LIN** (USART Frame Will be sent periodically every 50ms);

}