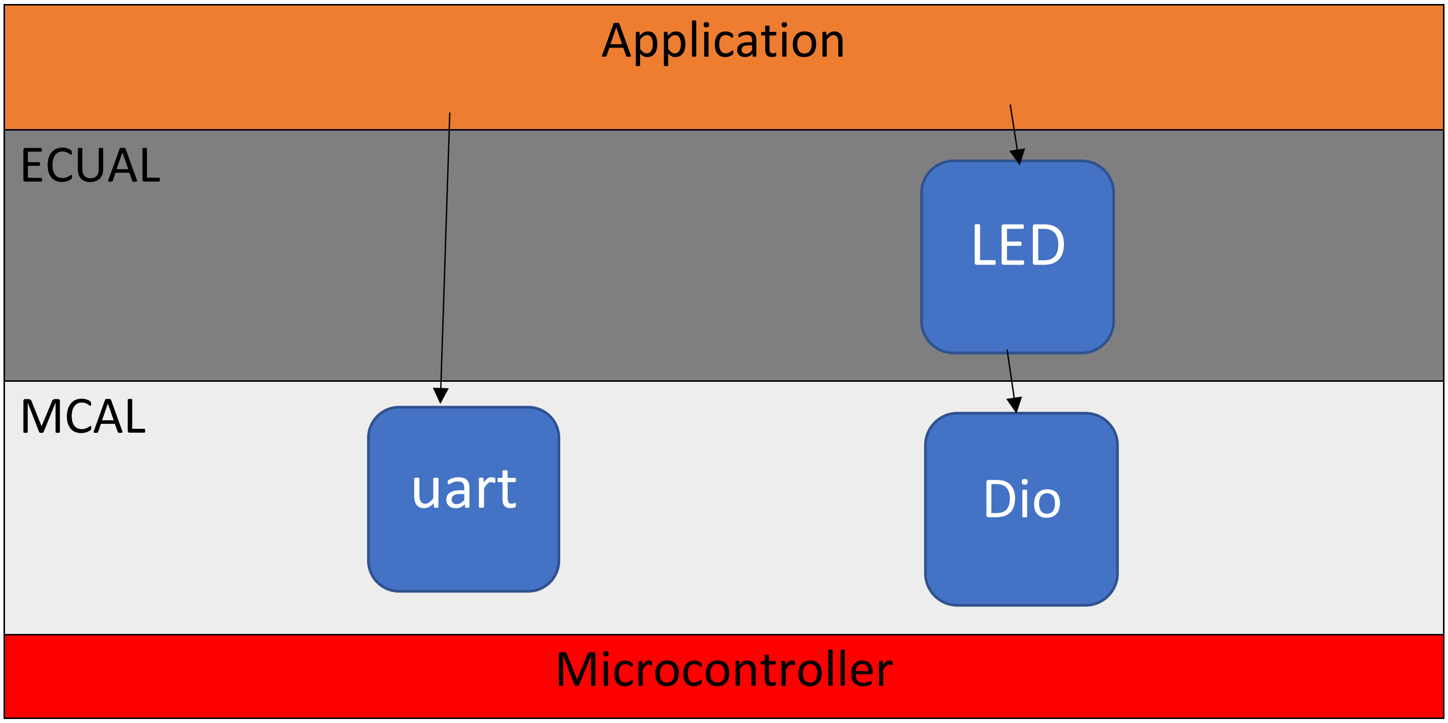
## Layers:



## DIO

```
enuDio_Status_t Dio_init(strDio_Config_t* strDio_pins);
enuDio_Status_t Dio_readPin(uint8_t u8_pinID, uint8_t *pu8_pinValue);
enuDio_Status_t Dio_writePin(uint8_t u8_pinID, uint8_t pu8_pinValue);
enuDio_Status_t Dio_togglePin(uint8_t u8_pinID);
```

# uart

```
UART_ERROR_t USART_Init(void);

UART_ERROR_t USART_Transmit(uint16_t u16_data);

UART_ERROR_t USART_Receive (uint16_t *Ptr_to_val);

UART_ERROR_t USART_Transmit_Packet(uint8_t *pu8_data,uint8_t SIZE);

UART_ERROR_t USART_receive_Packet(uint8_t *pu8_data,uint8_t u8_SIZE);

UART_ERROR_t USART_Transmit_INTDriven(uint8_t *pu8_data,void (*ptr)(void));

UART_ERROR_t USART_Receive_INTDriven(uint8_t *pu8_data,void (*ptr)(void));
```

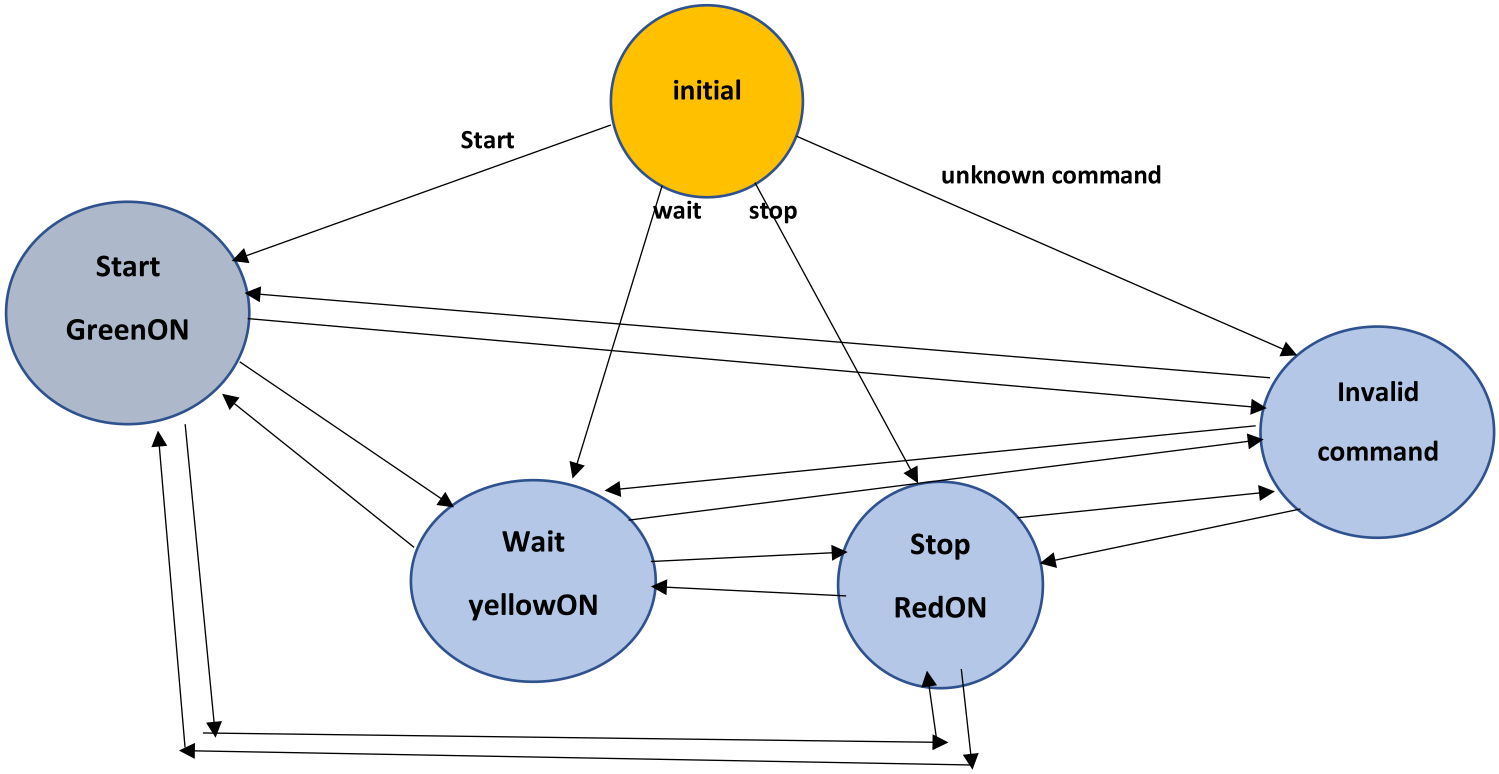
# LED

```
void ECUAL_LedInit(void);

void ECUAL_GreenSignON(void);

void ECUAL_GreenSignOFF(void);

void ECUAL_YellowSignON(void);

void ECUAL_YellowSignOFF(void);

void ECUAL_RedSignON(void);

void ECUAL_RedSignOFF(void);
```

# APP

```
APIs:

void APP_init(void);

void APP_GetState(void);

void APP_UpdateState(void);

Private functions:

void APP_receiveCommand(void);

enCompareState_t APP_StringsCompare(enStates_t Command_type)

void APP_sendResponse(uint8_t *ptr)
```

# State machine


initial
Start
wait
stop
unknown command
Start
GreenON
Invalid
command
Wait
yellowON
Stop
RedON