IBRAHEEM AL-NUAIMI
VT-18

Human-Machine and Electrical Interfaces

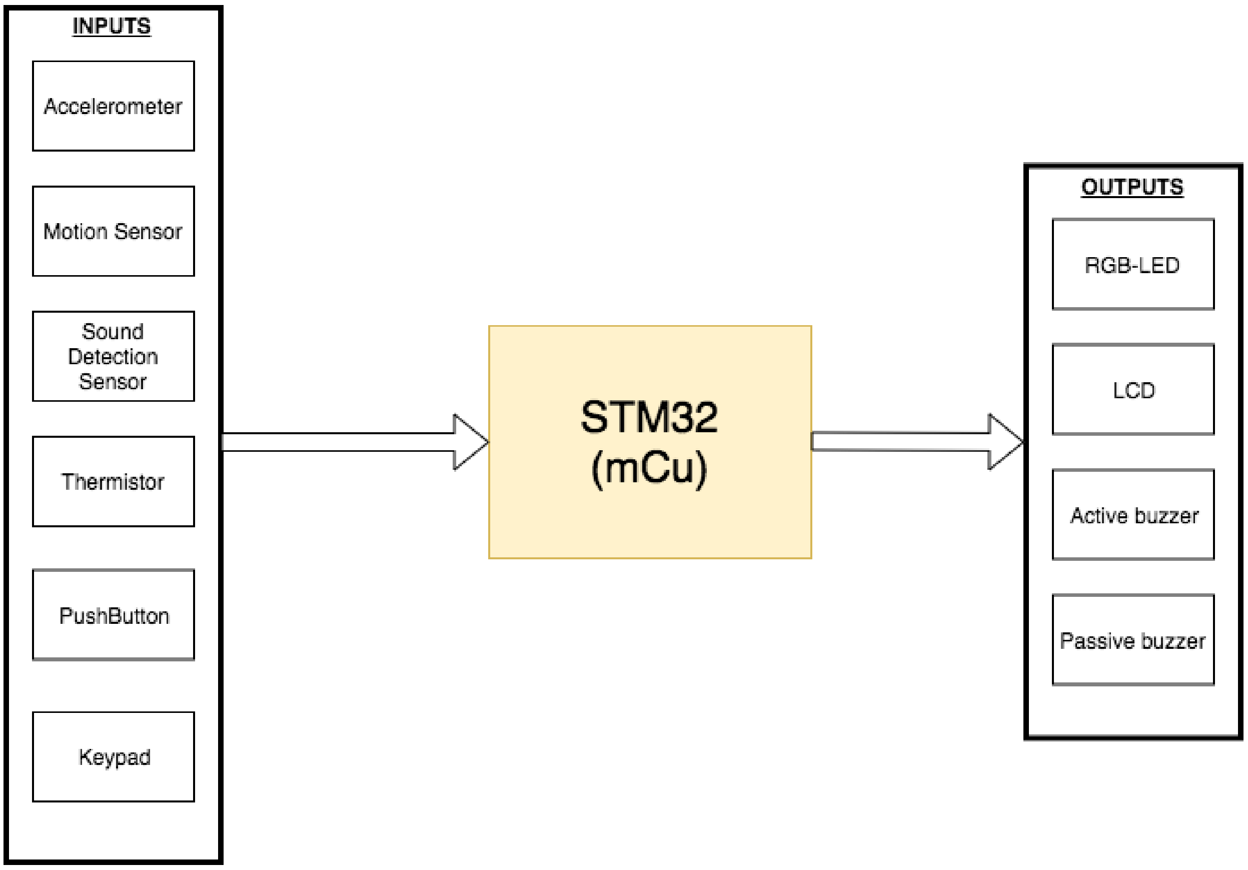
Background

Nordiska Museet has reached out to our team to solve an urgent problem. They have been presented the opportunity to host the largest diamond in the world. But unfortunately, their current alarm system is not up to par with the strict requirements set by the owner of the diamond. They must quickly upgrade their systems with a state-of-the-art alarm. None of the alarm systems available commercial-off-the-shelf (COTS) is good enough.

Project Goal

A high tech and advanced alarm system that will protect the diamond from everything.

OVERVIEW



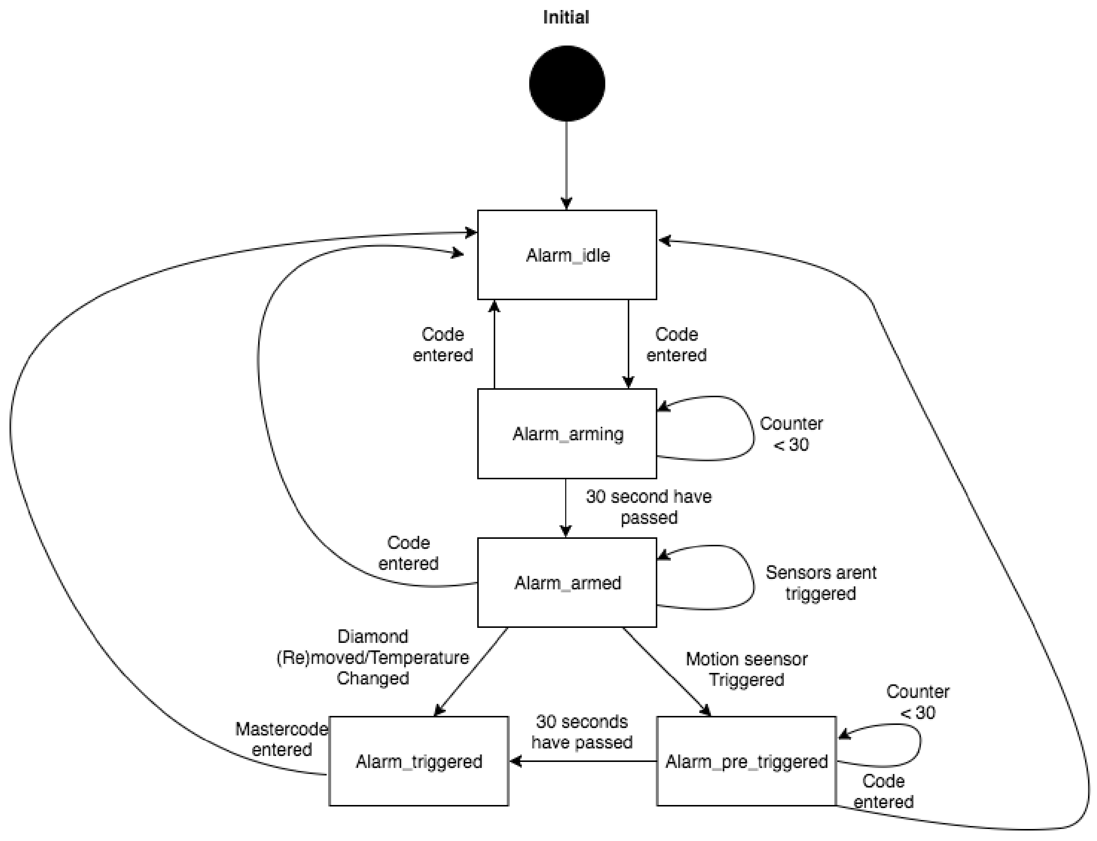
COMPONENTS

- STM32F401RE NUCLEO-64
- LCD 20X4
- Keypad 4X4
- PIR motion sensor
- RGB LED

Human-Machine and Electrical Interfaces

- ADXL345
- Sound detection sensor
- Pushbutton
- 2x breadboards
- YWROBOT POWER MB
- 6.5-12v PSU/ BATTERY
- 2x Buzzers
- Wires
- Thermistor
- Resistors

STATE MACHINE



FULFILLED REQUIREMENTS

- An alarm system with an access panel consisting of a display and a keypad. A user interface is shown on the display presenting the status of the alarm and the possibility to deactivate/active the alarm with a 4 digits long code.
- LEDS indicating the current state.
- A PIR motion sensor to detect any thermal activities.
- A mechanical switch integrated beneath the diamond.
- Two buzzers

- Temperature sensor is added to measure any temperature changes and will then compare it to the set temperature.
- The set temperature is programmable via the keypad in the idle state by pressing the ‘A’ key on the keypad.
- An accelerometer is added and used in the activity interrupt mode. The accelerometer will generate an interrupt indicating that an intruder is detected.
- The threshold level for the accelerometer can be programmed by pressing the ‘B’ key on the keypad.

ADDITIONAL FEATURES

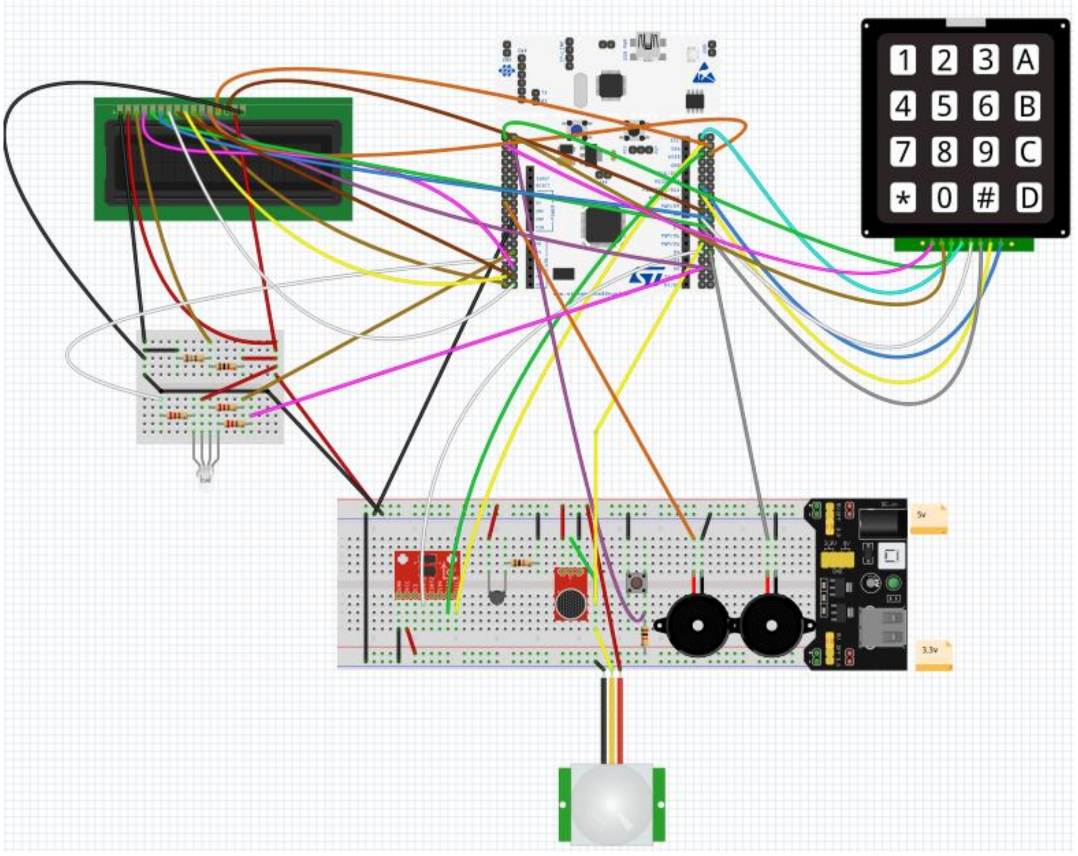
- Sound detection sensor
- The ‘D’ key works as a delete button when entering the code.
- The user code can be changed in idle state.
- A single RGB-LED instead of three LEDS.
- A bigger LCD-screen
- A cross-platform application displaying the actual state of the alarm and the temperature.

PINOUT & SCHEMATICS

PIN LABEL	PIN NUMBER	PIN DESCRIPTION	GPIO MODE
LCD_D0	PC0	LCD DATA PIN 0	OUTPUT
LCD_D1	PC1	LCD DATA PIN 1	OUTPUT
LCD_D2	PC2	LCD DATA PIN 2	OUTPUT
LCD_D3	PC3	LCD DATA PIN 3	OUTPUT
LCD_D4	PC4	LCD DATA PIN 4	OUTPUT
LCD_D5	PC5	LCD DATA PIN 5	OUTPUT
LCD_D6	PC6	LCD DATA PIN 6	OUTPUT
LCD_D7	PC7	LCD DATA PIN7	OUTPUT
LCD_RS	PB0	LCD RS PIN	OUTPUT
LCD_RW	PB1	LCD RW PIN	OUTPUT
LCD_E	PB2	LCD ENABLE PIN	OUTPUT
PAD_COLUMN_1	PA6	KEYPAD COLUMN 1	OUTPUT
PAD_COLUMN_2	PA7	KEYPAD COLUMN 2	OUTPUT
PAD_COLUMN_3	PA8	KEYPAD COLUMN 3	OUTPUT
PAD_COLUMN_4	PA9	KEYPAD COLUMN 4	OUTPUT
PAD_ROW_1	PC9	KEYPAD ROW 1	INPUT
PAD_ROW_2	PC10	KEYPAD ROW 2	INPUT
PAD_ROW_3	PC11	KEYPAD ROW 3	INPUT
PAD_ROW_4	PC12	KEYPAD ROW 4	INPUT

Human-Machine and Electrical Interfaces

ADC1_IN0	PA0	THERMISTOR	INPUT
LED_BLUE	PA10	RGB-LED BLUE PIN	OUTPUT
LED_RED	PA4	RGB-LED RED PIN	OUTPUT
LED_GREEN	PA1	RGB-LED GREEN PIN	OUTPUT
I2C1_SCL	PB8	SCL TO ADXL345	I2C
I2C_SDA	PB9	SDA TO ADXL345	I2C
GPIO_EXTI2	PD2	EXTERNAL INTERRUPT ADXL345	EXTI
GPIO_EXTI4	PB4	EXTERNAL INTERRUPT PIR/SOUND	EXTI
GPIO_EXTI5	PB5	EXTERNAL INTERRUPT BUTTON	EXTI
TIM2_CH1	PA15	BUZZER	PWM
ACTIVE_BUZZER	PB13	ACTIVE BUZZER	OUTPUT



CODE IMPLEMENTATION

The code is divided into a couple of header/source files. The main part of the program is implemented in the alarm source file.

Human-Machine and Electrical Interfaces

Enums defined in the alarm header file

Name	Values	Description
Alarm_state	Alarm_init, Alarm_idle, Alarm_arming, Alarm_armed, Alarm_PRE_Trigged, Alarm_Trigged, Alarm_SetTemp, Alarm_SetGyro, Alarm_setCode	Contains all the states for the state machine.
Key_Code	Key_No_Pressed, Key_Pressed, Key_OK,Key_Wrong, Key_A, Key_B, Key_C, Key_D	This enum is used for the function that returns the current state of the keypad. The function does also check if the right code is entered.
Sensor_status	NONE, Motion_Trigged, Sound_Trigged, Diamond_Trigged	This enum is used to indicate the states of the sensors. A variable of this type is declared and used together with the external interrupts to indicate the states of the sensors.
LED_COLOR	L_OFF, L_RED, L_YELLOW, L_GREEN	Contains all the different states the led can have in this program.
LCD_Status	LCD_NONE, LCD_Unlocked, LCD_Locked, LCD_Arming, LCD_PRI_Trigged, LCD_Trigged, LCD_SetTemp, LCD_SetGyro, LCD_SetCode	This enum is like the alarm enum and is used to control the printed text on the LCD.

Functions defined in alarm header file

Function type	Function name	Function parameter list	Function description
void	Alarm_status	void	The alarm state machine is implemented in this function and every state calls a function.
Alarm_state	A_idle	TextLCDType lcd, uint8_t setTemp, uint8_t code	This function is specifically for the idle state and relies on the function that checks the keypad status.

Human-Machine and Electrical Interfaces

Alarm_state	A_arming	TextLCDType lcd, uint8_t code	This function is specifically for the arming state and returns a different state either if the code is entered or 30 seconds have passed
Alarm_state	A_armed	TextLCDType lcd, uint8_t setTemp, uint8_t code	This function is specifically for the armed state and returns a different state if a sensor is trigged or the code is entered
Alarm_state	A_Pre_Trigged	TextLCDType lcd, uint8_t code	This function is for the pre_trigged state and contains a counter
Alarm_state	A_Trigged	TextLCDType lcd	If the user doesn't enter code when the alarm is set of will this function run and handle the Trigged state
Alarm_state	A_setTemp	TextLCDType lcd, uint8_t setTemp,	This function will run when the user enters the setTemp state and handles all the setups when the user want to change Temperature
Alarm_state	A_setGyro	TextLCDType lcd	Works like A_setTemp but changes the threshold of the gyro instead
Alarm_state	A_setCode	TextLCDType lcd, uint8_t code	Works like A_setTemp and A_setGyro but changes the code instead
Key_code	Alarm_code_status	uint8_t code	Handles the code input
void	update_lcd	TextLCDType lcd, LCD_status tmpS	Used to update the text print out on the LCD
void	lcd_clearRow	TextLCDType lcd, uint8_t row	Clears a row on the LCD
uint8_t	check_sensors	uint8_t setTemp	Checks the status of the sensors and temperature
int16_t	Read_analog_Temp	void	Reads the Analog value of the thermistor and converts it to a temperature in Celsius
void	Toggle_Buzzer	void	Handles the passive buzzer
void	set_Led	Led_Color ld	Used to set the color of the RGB led

Human-Machine and Electrical Interfaces

Functions defined in the keypad files

Function Type	Function name	Parameter list	Description
void	Keypad_init	void	Set all the keypad pins to high
Int8_t	Keypad_read	void	Checks if a button on the keypad is pressed by setting one column at a time and then checks the input from the rows. Returns 99 if a button isn't pressed
void	keypad_setColumn	uint8_t column	Set all the columns outputs to high and then resets a specific one.
uint8_t	Keypad_Checkrow	uint8_t column	Checks the rows input value and returns the pressed keys value if a key is pressed.

Functions defined in ADXL345 files

Function type	Function name	Parameter list	Description
void	ADXL345_writeRegister	uint8_t regst, uint8_t value	This function is used to write a value to a specific register
void	ADXL345_ReadRegister	uint8_t regst, uint8_t *data	This function will read from a specific register
void	ADXL345_ReadRegisters	uint8_t regst, int8_t *tmpData, uint8_t dataLength	Almost same as "..readregister" but reads from multiple registers
void	ADXL345_init	void	Initiates the gyro in measure mode.
void	ADXL345_init_interrupt	void	Initiates the gyro in interrupt mode
void	ADXL345_clear	void	Used to clear an interrupt
void	ADXL345_setThreshold	uint8_t threshold	Used to set the threshold