Magnetic Measurement System

Team Members:

Kamil Śladowski
Maciej Wiercioch

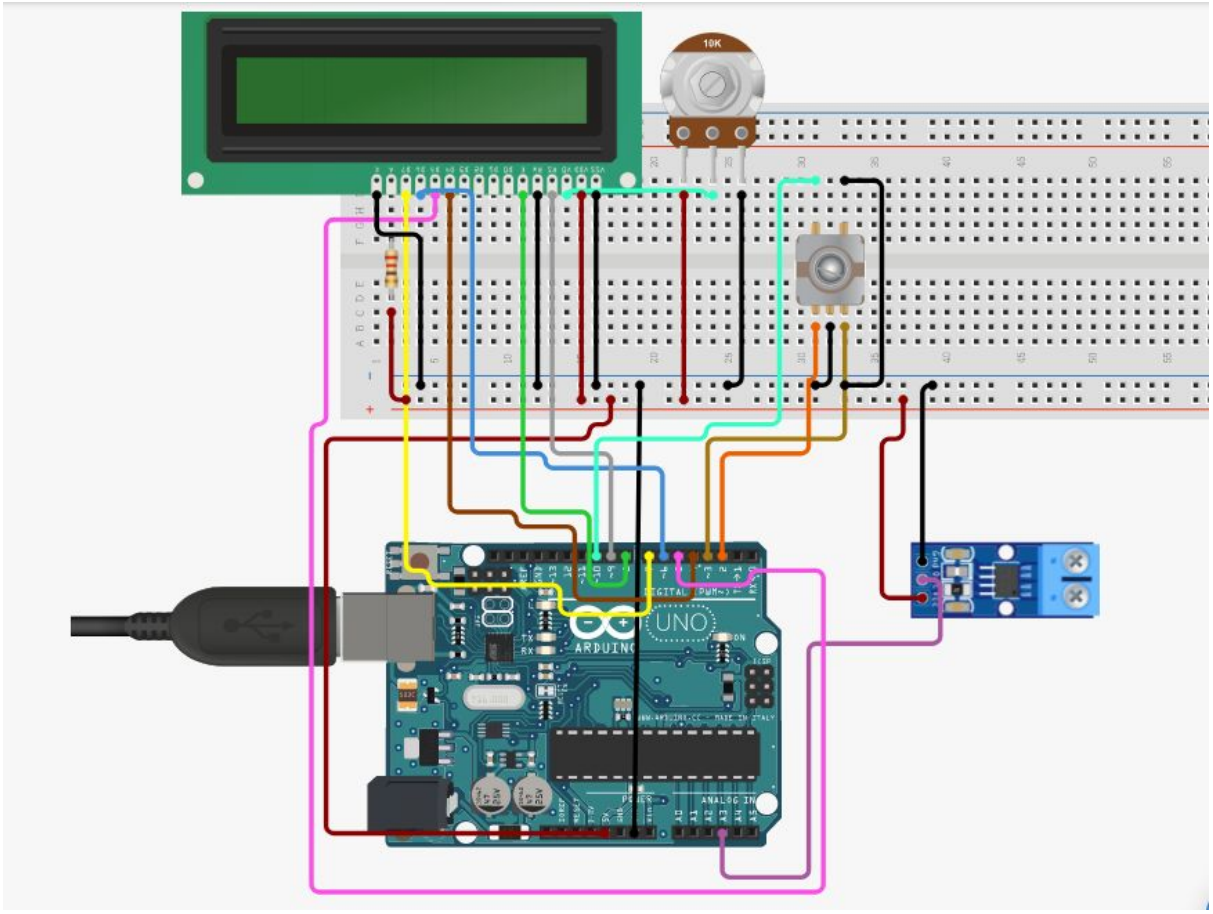
1. Overview.

Magnetic measurement system consisted of Arduino Uno, LCD display, magnetic sensor and rotary sensor was developed for saving measurements about magnetic field. Rotary sensor is used for changing time visible on LCD display.

Hardware required:

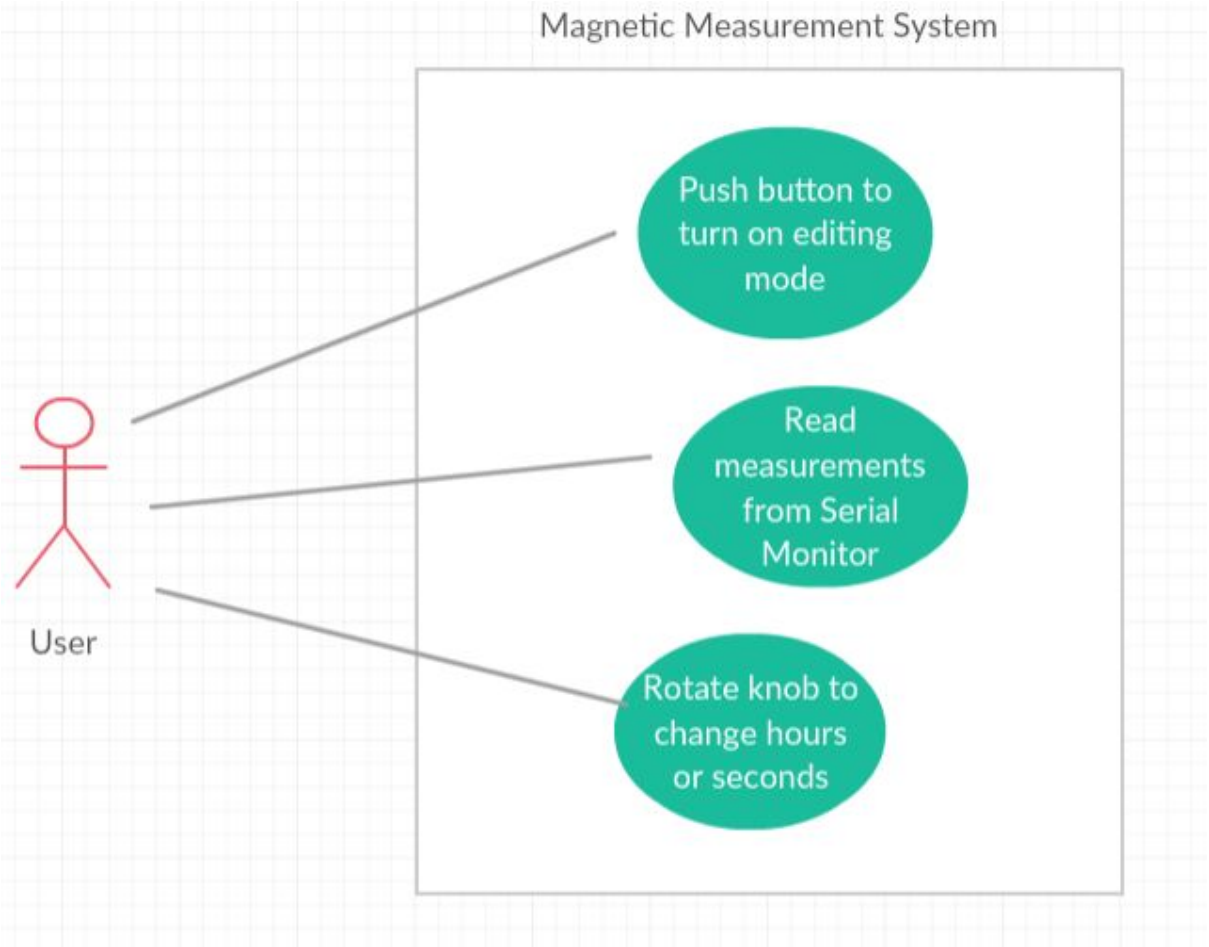
Arduino UNO Board
16x2 LCD Screen
Rotary Encoder Module
Hall Magnetic Sensor
Breadboard
Jumper Wires
10K Potentiometer

2. Schematics

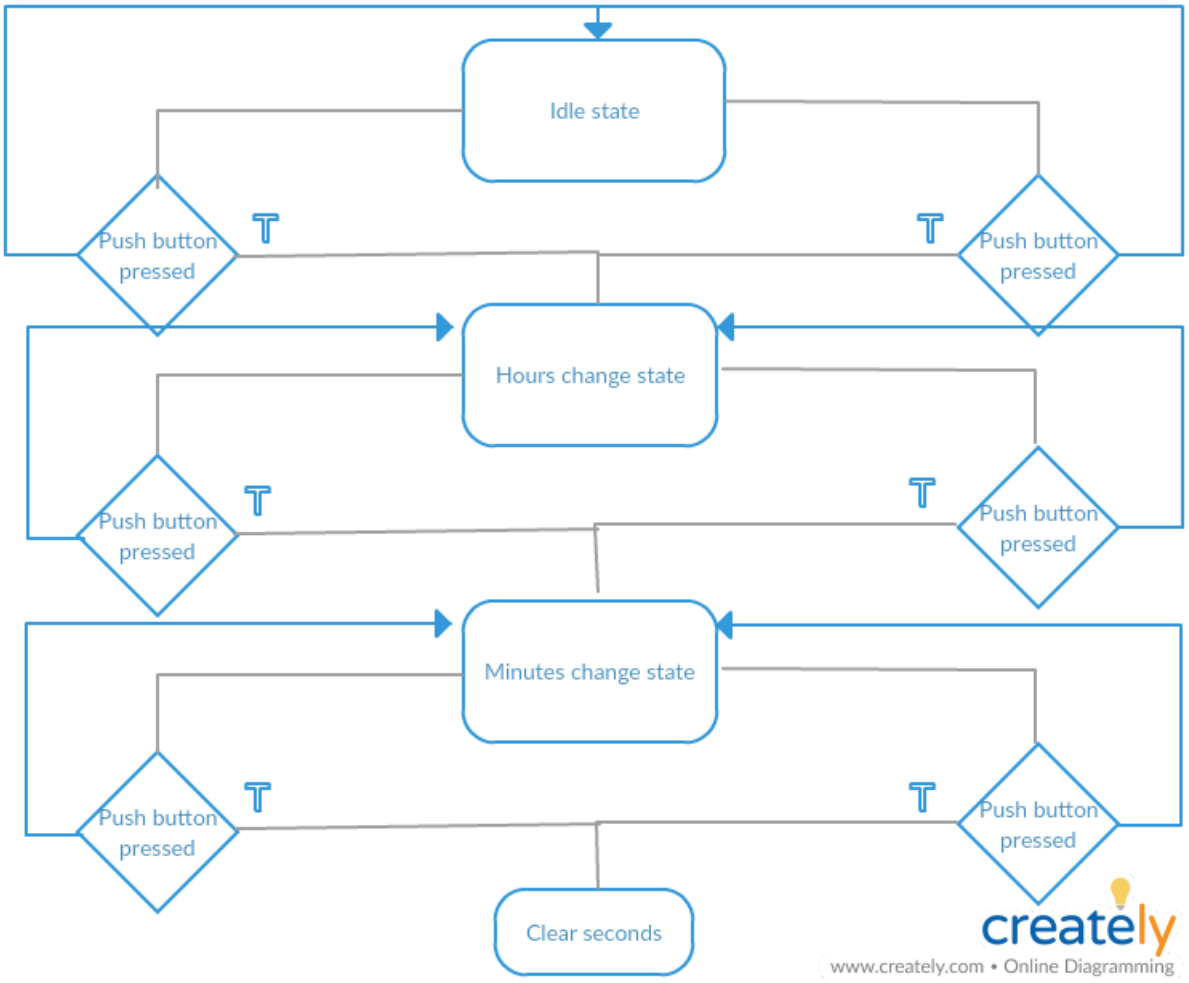


3. UML diagrams

Use Case Diagram



Activity Diagram



4. Functionalities

Functions

Function	Description
`void setup()`	Sets necessary pins, the data rate and prepares clock.
`prepareClock()`	Sets timer interrupts.
`ISR(TIMER1_COMPA_vect)`	Calls increment_time, read_measurements, manage_states and display_on_watch methods.
`void clearSeconds()`	Sets seconds to 0.
`void changeState()`	Changes actual system state.
`void clearState()`	Sets state to 0.
`void isInIdleState()`	Checks if system is in idle state.
`void getRotation()`	Returns 1 if right rotation occured and 2 if left occured. Otherwise returns 0.
`int isRightRotation(in aState, int bState)`	Checks if right rotation occured.
`int isPushButtonPressed()`	Checks if push button was pressed.
`void isPushButtonPressedTwice()`	Checks if push button was pressed twice.
`void increment_hours()`	Changes hours by one. The result depends on the rotation direction
`void increment_minutes()`	Changes minutes by one. The result depends on the rotation direction.
`void clearSecondsIfAppropiateState()`	Sets seconds to 0 if system is in second state.
`void increment_time()`	Increment global time.
`void read_measurement(String timestamp)`	Reads measurement value from the analog pin and prints it to the serial port.
`String get_blinking_timestamp(int h, int m, int s)`	Returns blinking timestamp.
`String normal_timestamp(int h, int m, int s)`	Returns timestamp.

`int* format_time()`	Returns formatted time in hh:mm:ss format using array.
`String get_timestamp()`	Returns blinking or normal timestamp.
`String format_digits(int i)`	Returns string consisted of two digits.
`void display_on_watch(String timestamp)`	Displays timestamp on LCD.
`void manage_states()`	Checks if system is in idle state, rotation occured or if button was pressed and depending on the event decide to call appropriate functions.

Known Issues:

- **The push button on rotary knob should be pressed for more than one second, to enter changing time mode;**
- **The logic of changing the minutes and hours using rotary knob does not work.**