

Low Level Speed Conversion Algorithm

Function Prototypes:

1. Initialize void function OLED_initialize with no parameters.
2. Initialize void function send_bit with the following parameters:
 - a. Pointer to GPIO_TypeDef.
 - b. Unsigned 32 bit integer pin_number.
 - c. Boolean variable bit_state.
3. Initialize void function OLED_sendByte with character data type argument.
4. Initialize void function OLED_sendCharacter with character data type argument.
5. Initialize void function OLED_sendCommand with character data type argument.
6. Initialize void function OLED_commandMode with no parameters.
7. Initialize void function OLED_setWrite with no parameters.
8. Initialize void function OLED_characterMode with no parameters.
9. Initialize void function OLED_enable with no parameters.
10. Initialize void function OLED_sendString with pointer to char data type.
11. Initialize void function OLED_sendInt with parameter integer to be displayed.
12. Initialize integer function calculate_speed with parameters being unsigned 16 integer period, and double data type numerator.
13. Initialize integer function send_speed with integer speed as its function argument.

Global Variable Declarations:

1. Unsigned 16 bit integer period, set to zero.
2. Integer revolution, set to zero.
3. Integer refresh_rate, set to 4.

Callback Functions:

- Note, the following function is called by HAL_TIM_IRQHandler() upon an input event.

HAL_TIM_IC_CaptureCallback(Timer_typedef * htim)

1. If the event is from TIM1, do the following:
 - a. Increment revolution until refresh_rate value is reached.
 - b. Reset the revolution to 1 if revolution is equal to refresh_rate.
 - c. Set period equal to return value of
__HAL_TIM_GET_COMPARE(&htim, TIM_CHANNEL_1).
 - d. Reset counter back to zero with __HAL_TIM_SET_COUNTER(&htim, 0);

main()

- Only including code I wrote. None of the code generated by cubeMX will be shown here.
1. Call OLED_initialize.
 2. Call HAL_TIM_BASE_Start(&htim2).
 3. Call HAL_TIM_IC_START_IT(&htim1, TIM_CHANNEL_1)
 4. Set numerator to $2\pi r \cdot 720$. Where r is the radius in centimeters.
 5. Set integer speed, boolean zero_flag, and 16 bit unsigned integer count_track to zero.
 6. In an infinite loop:
 - a. While we are in speed mode:

- i. Set count_track equal to return value of
__HAL_TIM_GET_COUNTER(&htim1).
 - ii. If revolution is equal to refresh_rate:
 1. Set speed to zero.
 2. Set speed equal to return value of
calculate_speed(period,numerator).
 3. Set the period to zero.
 4. Set zero_flag to one.
 5. Set revolution to zero.
 - iii. Else, if count_track is greater than threshold(TBD) and zero_flag is equal
to one, do the following:
 1. Set integer to_display to zero.
 2. Call function send_speed(to_display).
 3. Set zero_flag back to zero.
- b. End while loop.

OLED_initialize()

1. Call LCD_sendCommand(0x1).
2. Call LCD_sendCommand(0x38).
3. Call LCD_sendCommand(0x0E).
4. Call LCD_sendCommand(0x06).
5. Call LCD_sendCommand(0x17).
6. Call LCD_sendCommand(0x80).
7. Call LCD_sendString("Speed").

send_bit(GPIO_typeDef *port, uint32 pin_number, bool bit_state)

1. If bit_state == 1:
 - a. Assign port -> BSRR the equivalent bitwise or with itself and pin_number.
2. Else if bit_state == 0:
 - a. Assign port -> BRR the equivalent bitwise or with itself and pin_number.

OLED_sendByte(char character)

1. Call send_bit(D0_Port,D0_Pin, character & 1).
2. Call send_bit(D1_Port,D1_Pin, character & 2).
3. Call send_bit(D2_Port,D2_Pin, character & 4).
4. Call send_bit(D3_Port,D3_Pin, character & 8).
5. Call send_bit(D4_Port,D4_Pin, character & 16).
6. Call send_bit(D5_Port,D5_Pin, character & 32).
7. Call send_bit(D6_Port,D6_Pin, character & 64).
8. Call send_bit(D7_Port,D7_Pin, character & 128).
9. Call HAL_delay. Value passed in can be 1-10 (milliseconds).
10. Call send_bit(EN_Port,EN_pin,0)

OLED_sendCharacter(char character)

1. Call OLED_setWrite().
2. Call OLED_characterMode().
3. Call OLED_enable.
4. Call OLED_sendByte(character).

OLED_sendCommand(char character)

5. Call OLED_setWrite().
6. Call OLED_commandMode().
7. Call OLED_enable.
8. Call OLED_sendByte(character).

OLED_commandMode()

1. Call send_bit(RS_Port,RS_Pin,0).

OLED_setWrite()

1. Call send_bit(RW_Port,RW_Pin,0).

OLED_characterMode()

1. Call send_bit(RS_Port,RS_Pin,1).

OLED_enable()

1. Call HAL_Delay(). Value can be 5-10 (milliseconds).
2. Call send_bit(EN_Port,EN_Pin,1).

OLED_sendString(char * string)

1. While we haven't reached the end of the string:
 - a. Call OLED_sendCharacter(*string++).

OLED_sendInt(int to_display)

1. Initialize char array of size 10 stringNumber.
2. Call sprintf(stringNumber,integer value,to_display).
3. Call LCD_sendString(stringNumber).

send_speed(int speed)

1. Call OLED_sendCommand(0x1).
2. Call OLED_sendCommand(0x80).
3. Call OLED_sendString("Speed:").
4. Call OLED_sendCommand(0xC0).
5. Call OLED_sendInt(speed).
6. Return 1.

Andrew Capatina

11/26/2017

Version 2.0

calculate_speed(uint16 period, double numerator)

1. Set double speed to zero.
2. Set speed equal to numerator divided by period.
3. Typecast speed to an integer, add .5 for rounding. Set equal to speed.
4. Return speed.