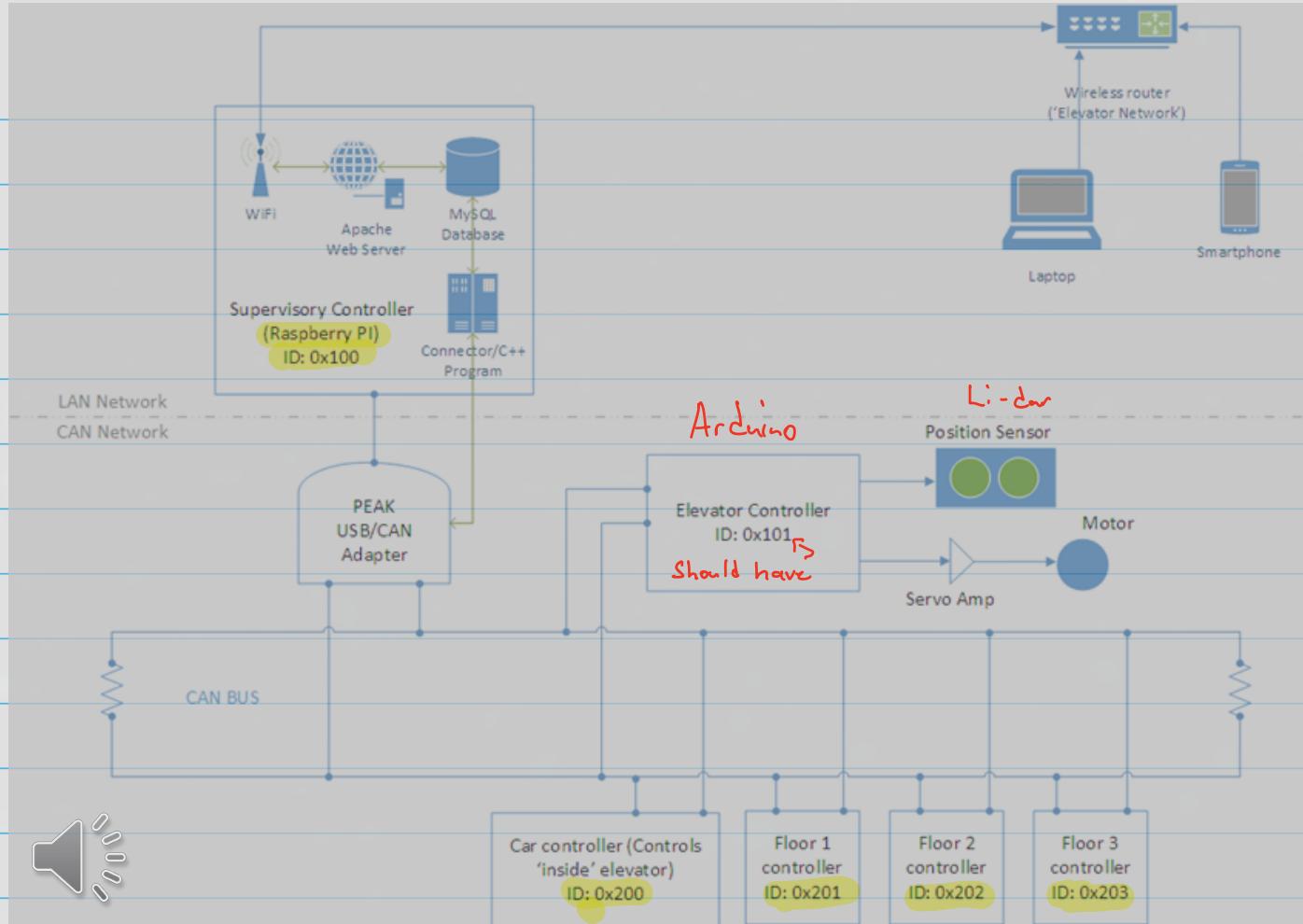


CAN / LAN Network Diagram



May 20th

Brandon Hawk

Set the ID for each as shown above in each respective STM32

↳ This can be set on line 34 in main.c

Current CAN Code

$T_x \rightarrow (AN \sim) R_x$

Blue Button Pushed \rightarrow GO_TO_FLOOR_1 (osrs) \rightarrow LED 2 illuminates 2 seconds.

T_x

Callback function - called by the interrupt, changes BUTTON variable.

CAN filter defined in STM32F3xx_hal_can.h, review last CAN slides for config!

Interrupt trigger calls EXTI15_10_IRQHandler() in STM32F3xx_it.c

↳ Chain linked to HAL_GPIO_EXTI_Callback() function

Line 374 is where we put the code to handle button inputs!

CAN Code Configuration

PC13
Ext I13 in use on GPIOC_13 (with blue push button)

13 is taken

Ext I 10-15 Activated.

PAS = LED2

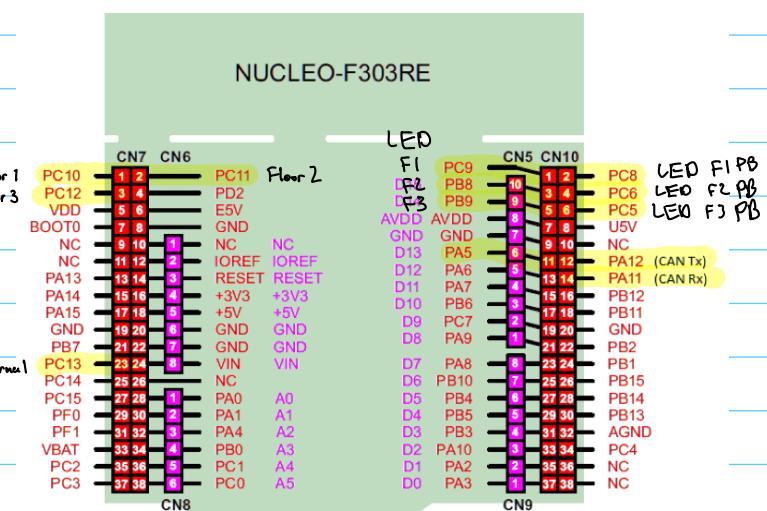
Hi output?
PC8 - Floor 1 PB LED
PC6 - Floor 2 PB LED
PC5 - Floor 3 PB LED

PC10 - Floor 1 Button
PC11 - Floor 2 Button
PC12 - Floor 3 Button

PC9 - Floor 1 LED

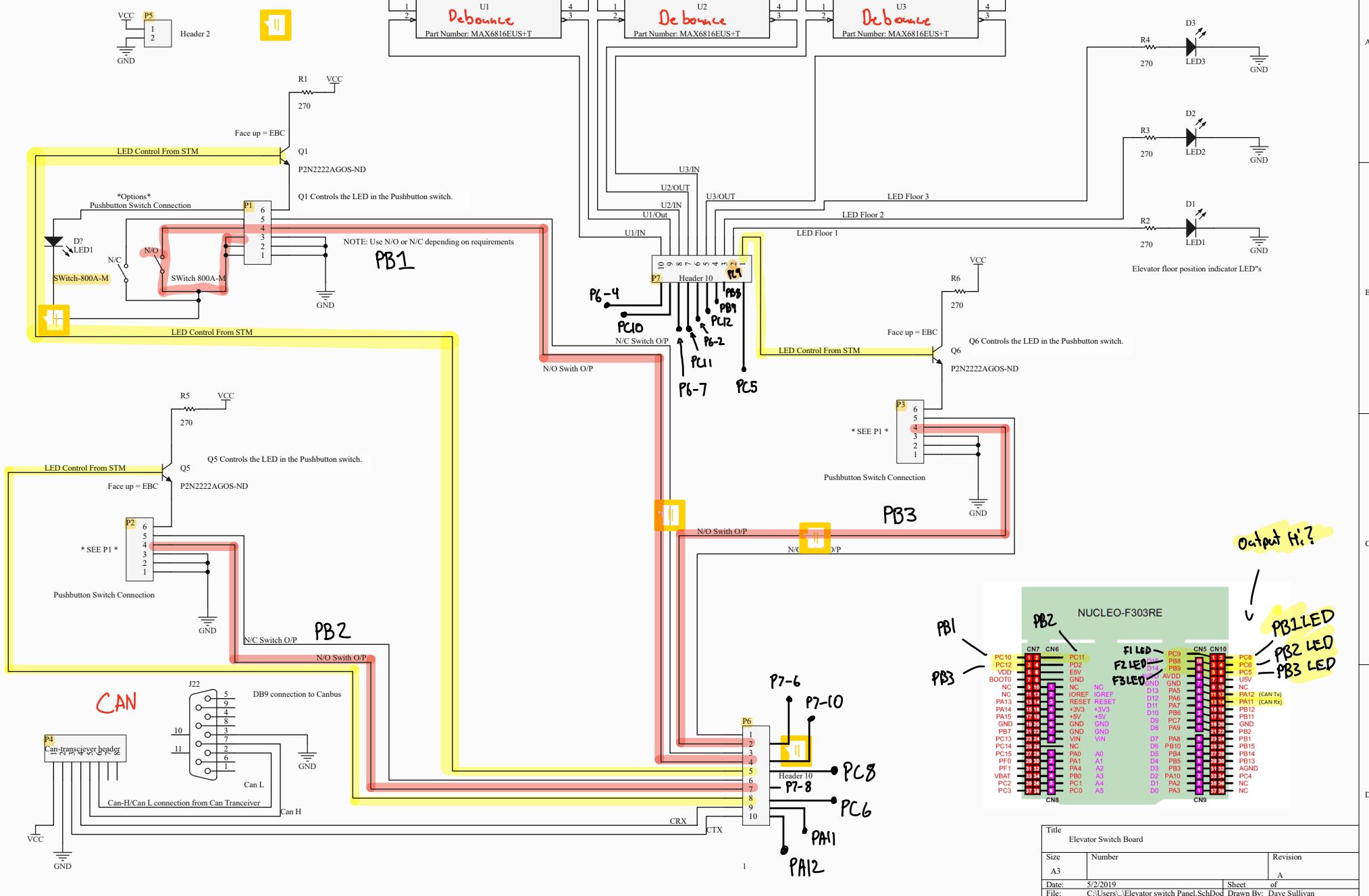
PB8 - Floor 2 LED

PB9 - Floor 3 LED

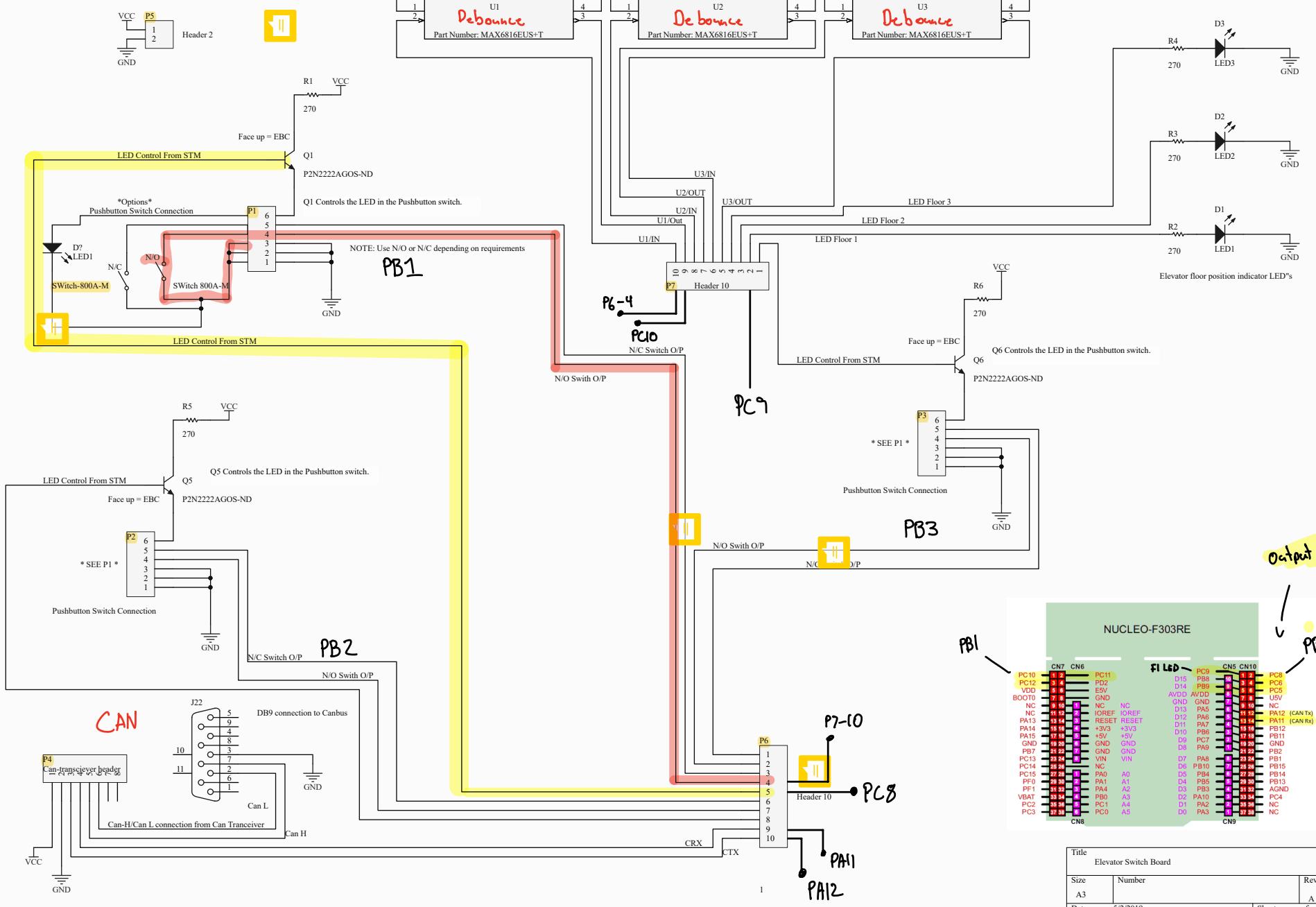


Switches will be used as NO

Wiring Diagram for Car Controller ID: 0x200

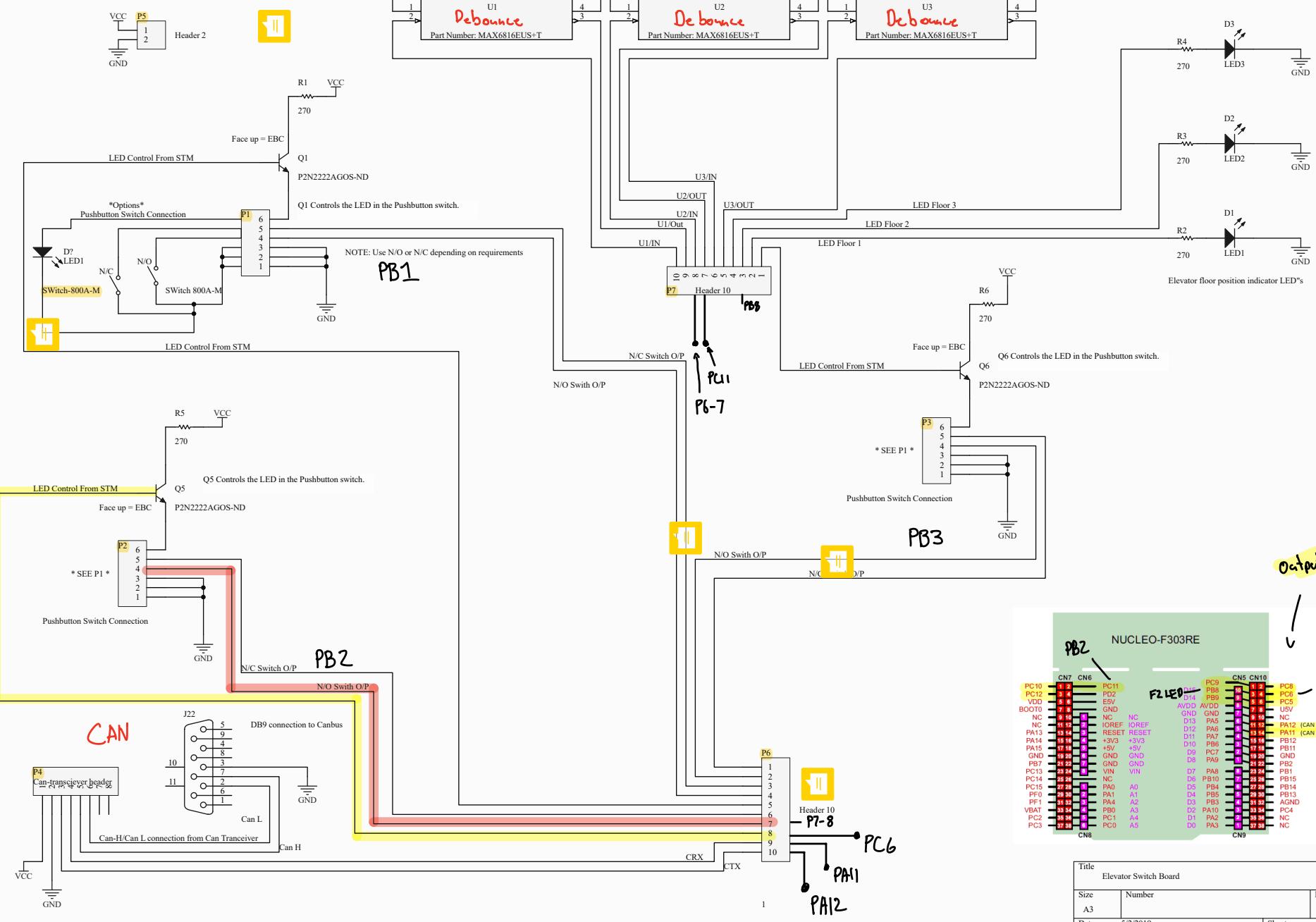


Wiring Diagram for floor 1 ID: 0x201



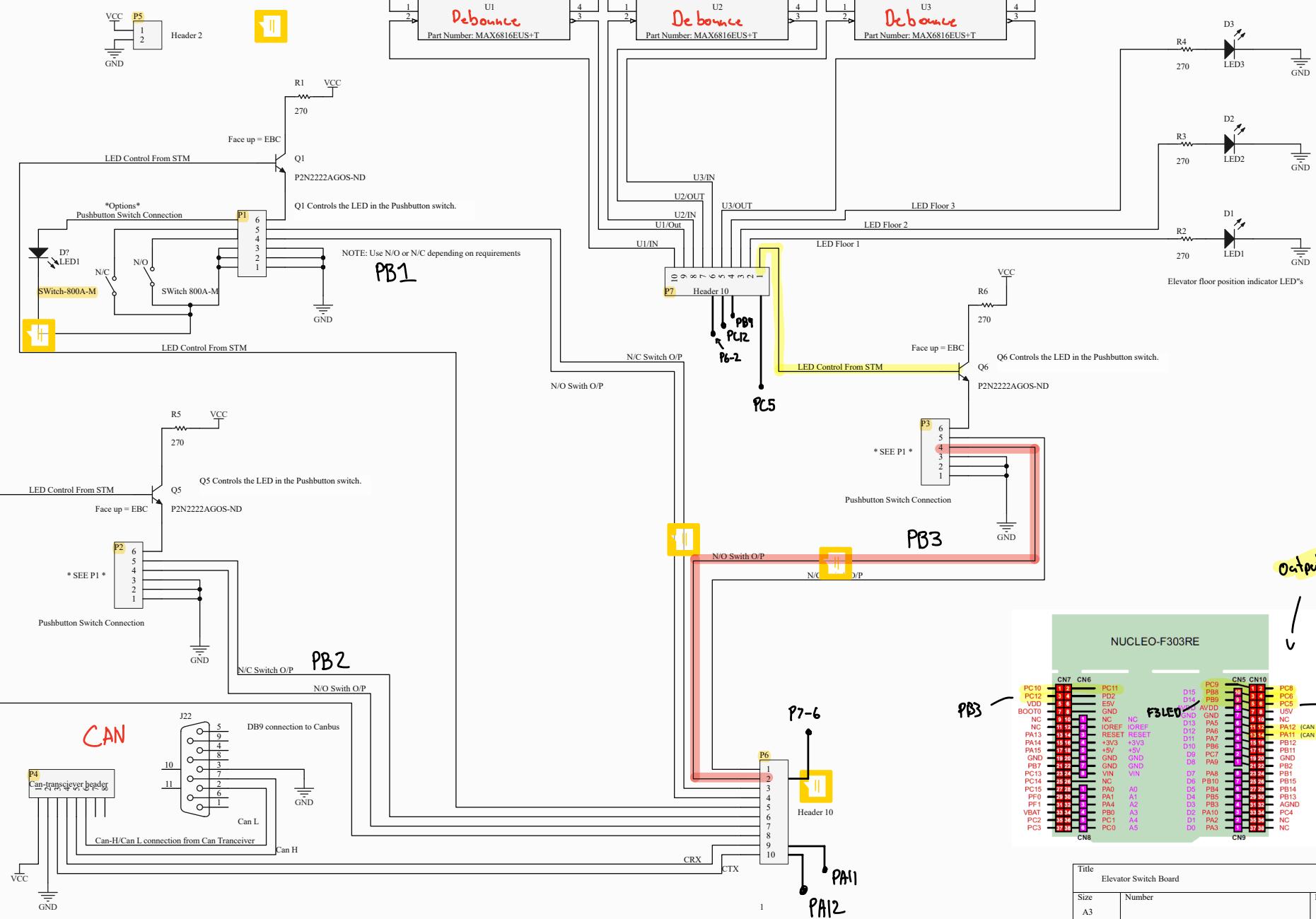
Title		Elevator Switch Board	Revision
Size	Number		A
A3			
Date:	5/2/2019	Sheet of	1
File:	C:\Users\...\Elevator switch Panel.SchDd	Drawn By:	Dave Sullivan

Wiring Diagram for floor 2 ID: 0x202



Title		Elevator Switch Board
Size	Number	Revision
A3		A
Date:	5/2/2019	Sheet of
File:	C:\Users\...\Elevator switch Panel.SchDd	Drawn By: Dave Sullivan

Wiring Diagram for floor 3 ID: 0x203



Title		Elevator Switch Board	Revision
Size	Number		
A3			A
Date:	5/2/2019	Sheet	of
File:	C:\Users\...\Elevator switch Panel.SchDoc	Drawn By:	Dave Sullivan

** Setting the ID and Mask Registers **

Receive Register looks like this ...

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STID[10:0]/EXID[28:18]												EXID[17:13]			
r	r	r	r	r	ID	r	r	r	r	r	r	r	5 bits	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXID[12:0]												IDE	RTR	Res	
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

Must Match up!

Both Filter ID and Mask Registers look like this ...

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
rw	rw	rw	rw	rw	ID / Mask	rw	<<	rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

To line up the Filter ID and Mask register bits with the Receive Register ID bits you must shift the bits in the Filter ID/mask registers by 5 when you set them

$$\begin{aligned} \text{Filter ID Register High} &= \text{ID} \ll 5 \\ \text{Mask Register High} &= \text{Mask} \ll 5 \end{aligned}$$

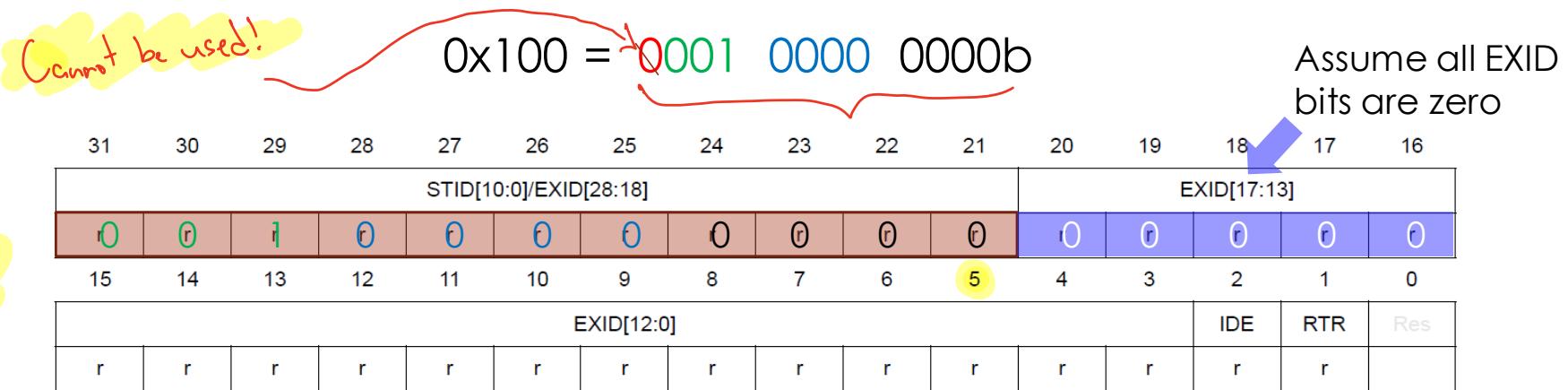
Filter ID/Mask
High bits

Filter ID/Mask
Low bits



Example: If ID = 0x100 what does the receive register look like?

There are only 11 bits in the Receive Register and we are storing a 12-bit hex number (i.e. 0x100) so the first bit is lost and the remaining bits are in the register



So, when we want to write a value to the Filter Register to compare with ID = 0x100 we must shift the ID bits $\ll 5$ so it lines up with the register

$$\text{Filter ID} = \text{ID} \ll 5$$

Similarly we must do the same thing for the Mask



Bit Arbitration
Priority

Setting up CAN Filters!

Supervisory = 0x100 =	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
Elevator Controller = 0x101 =	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
Car Controller = 0x200 =	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
Floor 1 Controller = 0x201 =	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
Floor 2 Controller = 0x202 =	0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0
Floor 3 Controller = 0x203 =	0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1
Filter Mask ID High = 0xEF C	1 1 1 0 1 1 1 1 1 1 1 0 0

Prevents 0x3xx from passing filter!

for Mask : 1 = Must Match 0 = Don't care

```
326 //Set up CAN Rx Filters
327 CAN_FilterTypeDef filter;
328
329 //Configure filter 0 to direct everything to FIFO 0
330 filter.FilterBank = 0;
331 filter.FilterIdHigh = ID << 5;
332 filter.FilterIdLow = 0x0000;
333 filter.FilterMaskIdHigh = 0xEFC << 5;      //Only allows 0x10x or 0x20x ID's to pass filter
334 filter.FilterMaskIdLow = 0x0000;
335 filter.FilterFIFOAssignment = CAN_FILTER_FIFO0;
336 filter.FilterMode = CAN_FILTERMODE_IDMASK;
337 filter.FilterScale = CAN_FILTERSCALE_32BIT;
338 filter.FilterActivation = ENABLE;
339 filter.SlaveStartFilterBank = 0;
340
```