**CAL - Initial plan for assembly project**

1. Put a timer (before starting and between two sequence)
2. Find how to turn on a led randomly
3. Use register to save the randomly sequence
4. Make a real-time comparison between the randomly sequence and user input
5. Make a way for the user to see if he made a mistake
6. If wrong input then try again (number of life) (how many time he can retry, how many live left)
7. Make a way to see if input correctly the sequence
8. Make a way to see when the game ends.
9. Give him 10000000 kr.
10. For the timer we can use this delay function => 
11. Turn on a led randomly..

There is \_ leds on the STK600 ..

How it will works

Turn on the STK 600

Press any button to start

Wait few seconds

A led blinking randomly (or we can choose a sequence I don’t know if it’s possible)

Wait until the user press a button (his choice)

User have to choose where the button blinked

If good answer

wait

(Next turn)The 1st led will blink again and a new led will blink (it can be the same led)

Wait until the user press a button (his choice)

User have to choose where the button blinked first and the next one..

If good answer (Next turn) 3rd led will.. etc..

If wrong answer

Make blinking all led 2rd time to show he made a mistake

Wait

Show the correct path from the beginning \_ times (number of life)

If game over (all the led blink for 5 sec for example) and it’s finish

If win make the led blinking 5 times or turn on turn off each led one by one.

Function & variable needed

Delay / Timer

Variable to store what the user press (array)

Variable to store which led blinked (array)

Variable to inc the number of led

Variable to dec the number of lives

Function good answer

Function bad answer

Function mistake

Function game over

Function game win.

Push a value

Add a delay max 1 sec

Press a button

Compare the switch w/ stack

If correct … blinking thing

and next // how many led will blink until the end of the game ?

If false blinking thing =>

pop the value (no retry) and restart