Luciebox

Owners manual

The Luciebox pretbak, made by Uncle Lode Ameije for Lucie.
Project started 2019-04-02. Manual v20210406

Introduction

I saw a toddler being mesmerized with the sounds and lights after pressing random buttons on one of my multiTimers. Right away I knew what I should give to my brother's first soon-to-be-born child. What started as a weekend project now has a 40 something pages manual, and Lucie is two years old. The Luciebox is not going to appeal to many people, but I know, if I would have had this box as a kid, I would have worn it out together with my brother. We'd know every feature and bug better than the original creator. Sure, I can claim that there is educational value in this over the top retro device. But, let's not forget the original mission of this project… It's time to mesmerize! As I realize now there is no such thing as finished projects, only abandoned projects, I'm welcoming all suggestions and contributions. Looking at you Lucie! Regards from your high-expectations-uncle Lode.

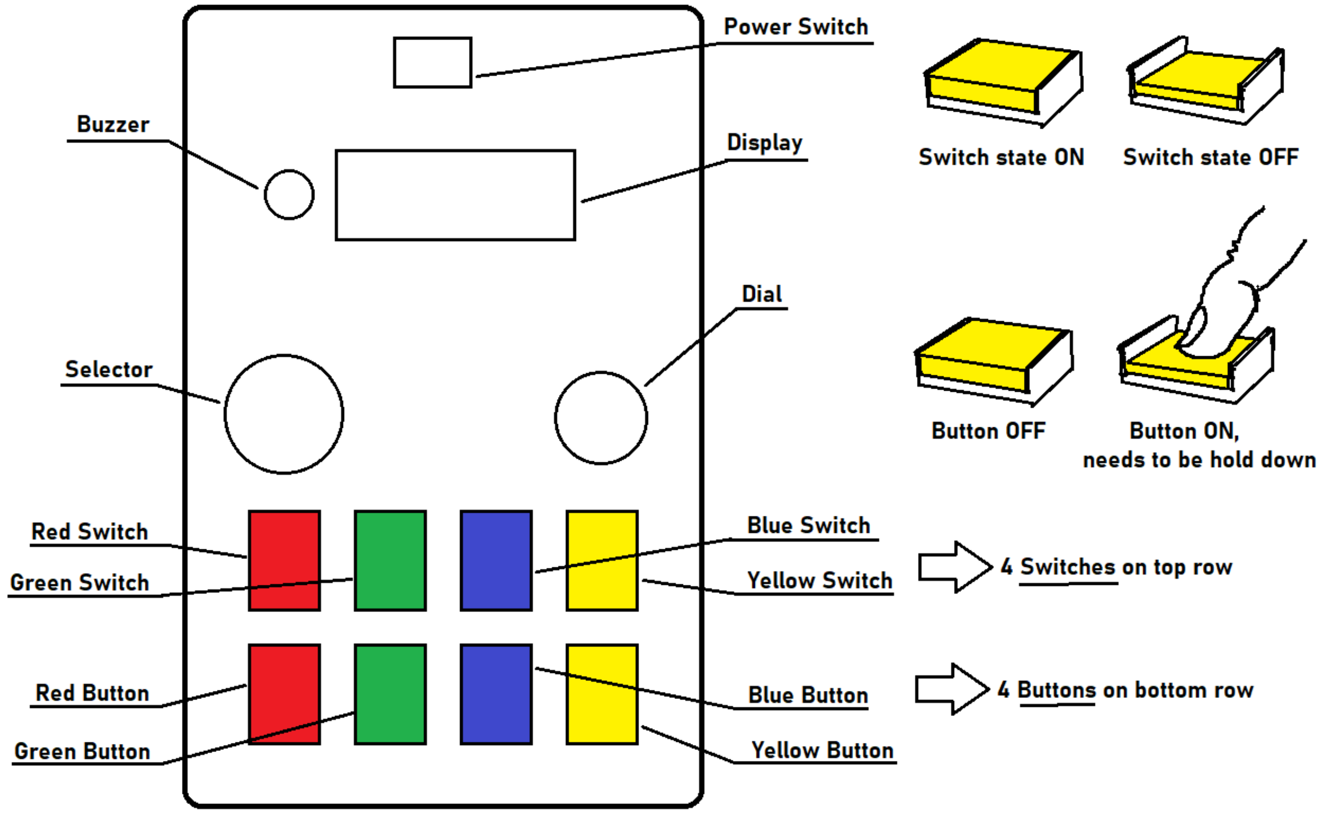
How to use this box

Don't be put off by the lack of information on the box. It's ok to be overwhelmed. Just know that there are 23 applications. An app is selected with the selector dial in combination with the Red Switch. Applications have different degrees of difficulty. Choose an application from the list and try to make it work. I suggest Guitar Hero for a fun introduction. Read the instructions for the specific app and have a go at it. At start leave all the switches in the OFF position to enable the default mode. Experiment away, you can't do anything wrong.

Table of contents

Getting started

Overview



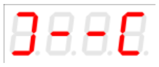
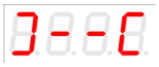
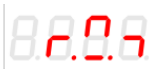
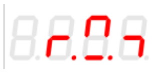
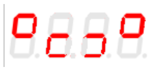
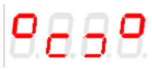
Name in this document	Official name	Description
Power Switch	On / Off switch	Switches on the box with the batteries. If powered by USB or the programming connecter, this will have no effect
Selector	Selector dial	Limitless 12 positions dial, on the left of the lid
Dial	Encoder dial	Limitless smooth dial on the right of the lid
Red Switch	Latching Button 0	Red toggle button which can be set in the ON or OFF position.
Green Switch	Latching Button 1	Green toggle button which can be set in the ON or OFF position.
Blue Switch	Latching Button 2	Blue toggle button which can be set in the ON or OFF position.
Yellow Switch	Latching Button 3	Yellow toggle button which can be set in the ON or OFF position.

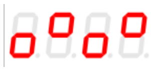
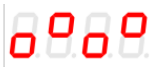
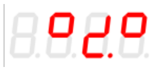
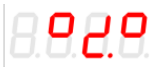
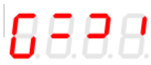
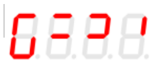
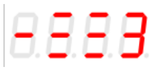
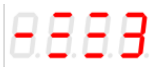
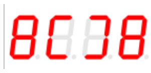
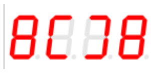
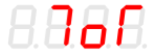
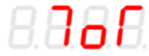
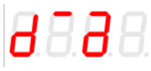
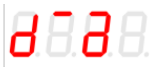
Red Button	Momentary Button 0	Red push button. When pushed and held, it’s ON, when released, it’s OFF.
Green Button	Momentary Button 1	Green push button. When pushed and held, it’s ON, when released, it’s OFF.
Blue Button	Momentary Button 2	Blue push button. When pushed and held, it’s ON, when released, it’s OFF.
Yellow Button	Momentary Button 3	Yellow push button. When pushed and held, it’s ON, when released, it’s OFF.
Buzzer	Buzzer	A tiny speaker is inside the box. The volume can be set by putting some tape over the hole. The sound can be muted in the settings app
Display	4 digit 7 segment LED display	Every digit has actually eight segments if we include the little dot, called the decimal point, on its right

Applications

Overview

If one of the buttons is held while rotating the selector knob, the splash screen stays visible.

App	**Selector**	**Latching 0**	**Splash screen**	**comments**	**Difficulty [1-5]**
Settings	1	0		Test all the buttons. Check configuration.	1
	1	1			
Total Time	2	0		Display time from the moment the box is switched on	2
Stopwatch	2	1		All the goodies from a typical 80’s stopwatch. Has two independent timers.	2
MultiTimer	3	0		A game clock for up to 4 players. Can also be used as a chess clock or as a kitchen timer.	3
Pomodoro timer	3	1		The Pomodoro technique helps procrastinators to get things done	3

Random world	4	0		Roll the dice, flip a coin, play bingo, draw random playing cards or let psychic Lucie answer all your questions.	2
Score keeper	4	1		Keeps a tally of the score of up to four players.	2
Simon says	5	0		Try to repeat light patterns of ever increasing length. Has multiplayer modes.	3
Quizmaster, reaction game	5	1		Press your button as fast as possible when the lights go on. Up to four players.	3
Whack a mole	6	0		Hit the button as quick as possible if the light goes on. If you play with sounds, it's called "whack a bird"	2
Guitar hero	6	1		Bars are falling down quickly, hit the buttons at the right moment to catch them!	3
Sketch pad	7	0		Scroll through and edit 95 pictures	3
Memory	7	1		Reproduce what the display showed.	3
Radio	8	0		Play the main songs, your own compositions and the stored sequencer loops	1
Media player	8	1		A full featured media player. Combine the built in or stored animations with one of the songs. It could aswell have been named "my first meme".	3
Metronome	9	0		Set the number of beats per minute. The flash mode produces noiseless ticks.	2
Sequencer	9	1		4 bars of 8 notes of fun. Always starts up with an auto generated sequence.	4
Music notes	10	0		Experiment with tones, lengths, scales and arpeggios. Features an auto mode	1
Composer	10	1		Create and save up to four songs.	4

Geiger	11	0		Measure your local radio activity. Or generate all kinds of noise.	1
Hack mode	11	1		Display and edit the Luciebox memory. Experiment as you like.	5
Dream time	12	0		Baby soothing patterns and sounds.	1
Letters and Numbers	12	1		Learn how to count, learn the alphabet	2

Unimplemented apps

App	status	comments	Difficulty [1-5]
Slots machine	not started	Teach children that luck can replace work in order to make money	3
Timed key press game	not started	Olympics game with keypresses. Press as fast as possible, stay in a steady rhythm or gradually go faster.	4

Toddler

Radio

Simple music player. Each momentary button will play a song. The standard songs are featured, and your own programmed songs!

Button	Functionality	Comments
Latching_1	Song bank selector 1	A combination of the bank selector assigns different songs to the momentary buttons
Latching_2	Song bank selector 2	A combination of the bank selector assigns different songs to the momentary buttons
Latching_3	Playback speed / Transpose	Dial will change the settings while a song is playing

Songs

Latching button index		
2	**3**	**Functionality**
0	0	Permanent memory songs bank 1 assigned to momentary buttons
1	0	Permanent memory songs bank 2 assigned to momentary buttons
0	1	Home made songs from the song composer assigned to momentary buttons
1	1	Sequencer songs 1 to 4 assigned to momentary buttons

Music notes

Plays single notes. Manually or automatically. Test different scales and modes.

Button	Functionality [OFF / ON]	Comments
Latching_1	Simple mode / Advanced mode	see different chapters for controls in each mode
Latching_2	Single note / Queue	In queue mode, all pressed notes will be added to the buffer. In single mode, the buffer will be erased before adding a note.
Latching_3	Manual mode / Auto arpeggio	Auto arpeggio will automatically play the next note after a timed interval
Dial		in auto arpeggio: set interval, in manual mode: go up or down the scale

Simple mode

Button	Functionality	Comments
Momentary_0	Mute	Will not make sound while changing notes
Momentary_1	Play active note	Just play the last played note
Momentary_2	Previous note	Previous note on the active scale
Momentary_3	Next note	Next note on the active scale

Advanced mode

Button	Functionality	Comments
Momentary_0	Duration	Press to cycle between full, half, quarter, eight notes
Momentary_1	Progression	Press to cycle between

		- MANU: Manual (default, set direction yourself), - UP: Going up the scale, - DO: Going down the scale, - UPDO: Saw pattern, going up and down, changing at a random interval - RND: Random picking of a note on the scale - CRAY: Crazy mode going randomly up and down at erratic intervals.
Momentary_2	Scale	Cycle between different scales: - AJOR: Major - INOR: Minor - PENT: Pentatonic - BLUE: Blues Major - CHRO: Chromatic
Momentary_3	Key	Cycle between all 12 tones in an octave to set the key of the selected scale.

Metronome

A classic metronome with fun visual effect.
12 steps per cycle, up to three tickers.

Button	**Functionality [OFF / ON]**	**Comments**
Latching_1	Make ticker 2 step backwards	
Latching_2	Set speed in beats per minute (bpm)	One ‘beat’ is a full circle of 12 steps.
Latching_3	Manual mode / Ticking mode	
Momentary_0	Reset the positions of all tickers	Set to the position of ticker 1
Momentary_1	Increase ticker 2 offset by one	
Momentary_2	Increase ticker 3 offset by one	

Metronome Manual mode

Button Momentary_3	**Functionality**	**Comments**
Single press	manual step	
Not pressed	rotate dial to step	
Hold and dial	Change tick sounds	note length: shorter (rotate clockwise) or longer (rotate counter clockwise) beeps. Shorter beeps lose their pitch, and are more like a metronome ‘puck’ sound.

Metronome Ticking mode

Button	Functionality	Comments
Momentary_3	Flash screen toggle	Will toggle screen flashing at every beep. This improves metronome feedback in loud environments.
dial		change bpm

Dreamtime

Mesmerize and go on a trippy hypno voyage where sound and graphics are ruled by randomness and chaos.

Button	Functionality [OFF / ON]	Comments
Latching_1	Pitch offset	Offset is set with dial
Latching_2	Unmute / Mute	Every segment has a sound assigned.It's played when it appears or disappears. Sounds are a 32 values range off the total of 256 available sounds. The offset is settable.
Latching_3	Auto / Manual	Dial will change step in manual and set interval speed in auto mode
Momentary_2	Previous step	Only in manual mode
Momentary_3	Next step	Only in manual mode

Child

Easy timer

Displays the time since the box was switched on. There are no controls in this app.

Letters and Numbers

Learn how to count and learn the alphabet.
Learn how to be a bouncer by using the counting function to keep track of the number of guests in your bar.

Button	Functionality [OFF / ON]	Comments
Latching_1	Number mode / alphabet mode	display numbers or letters
Latching_2	Numbers as Decimal / Hexadecimal	Only effective in number mode
Latching_3	Manual / Auto count	Auto mode

		● Song is played when in alphabet mode ● Dial changes the speed Manual mode ● Dial changes value
Momentary_0	in auto mode: toggle count up / down	
Momentary_1	set value zero	
Momentary_2	value down	
Momentary_3	value up	

Score tracker

Keeps up to four different tallies. Can also be used for general counting. Cribbage, snooker, darts,...

Button	**Functionality [OFF / ON]**	**Comments**
Latching_1	Count up / Count down	
Latching_2	Apply score change to single tally / to all tallys	
Latching_3	Modify mode / View mode	In view mode, every tally can be checked without modifying it
Momentary_0	Tally 1	
Momentary_1	Tally 2	
Momentary_2	Tally 3	
Momentary_3	Tally 4	
dial	Hold momentary + dial → set delta	Will show the delta on the screen as long as the momentary button is not released.

Movement detector (tilt fun)

Deprecated app. Will be made available again on popular request and when mercury switches are deemed environmentally friendly.
Move the box around to count.

Button	**Functionality [OFF / ON]**	**Comments**
Latching_1	go through all four positions for a full cycle / skip lateral movements for a full cycle.	don't take lateral movement into account

Latching_2		
Latching_3		
Dial		

Move the box around, inside are motion detectors. When moved through a cycle correctly, the counter will increase.
Put the box on a swing, see how many times you can swing in a minute!

Sketch pad

All drawings are stored in eeprom. Scroll through and edit the drawings.

Modes

Button	**Functionality [OFF / ON]**	**Remark**
Latching_1	ON: Save menu	Save a drawing to the eeprom memory
Latching_2	Display Image / Display Index	“Display index” displays the eeprom drawing address
Latching_3	View / Draw	In draw mode, drawings can be edited

Display mode

Control	**Functionality**	**Screen**
Momentary_0		
Momentary_1	Global display setting	cycle through: ● normal (default) ● negative ● all on ● all off ● STOP screen
Momentary_2	Previous drawing	
Momentary_3	Next drawing	
Dial	Scroll through drawings	

Draw mode

Control	**Functionality**	**Screen**
Momentary_0	Toggle active segment	
Momentary_1	Global display setting	cycle through:

		● normal (default) ● negative ● all on ● all off ● STOP screen
Momentary_2	Move cursor inside digit	
Momentary_3	Move cursor to next digit	
Dial	Move cursor	

Save To eeprom Mode

When scrolling through images, will NOT update the screen. The buffer stays the same. This way, you can scroll through the desired image (used index to see number) and save .

Control	**Functionality**	**Comments**
Momentary_0	Save screen	Will save the drawing to the active EEPROM address (visible with Latching_1)
Momentary_1	HOLD to have “shift function for momenary_2 and momentary_3	
Momentary_2	previous drawing OR with SHIFT FUNCTION: delete drawing slot (and move all indexes,)	Hold down Momentary_1 for it to work
Momentary_3	next drawing OR with SHIFT FUNCTION: insert drawing slot and move all indexes from other drawings. Will overwrite the last drawing from eeprom.	Hold down Momentary_1 for it to work
Dial	move through drawings.	

How to create animations

Step	**Comment**	**Example**
1	Create and save a drawing with the STOP marker. (Draw mode → global display settings → stop screen)	drawing 27: STOP
2	use as many frames as you want the next drawings	drawing 28: UNCL drawing 29: LODE drawing 30: IS drawing 31: ~~OLD~~ COOL
3	To indicate that the drawings are an animation, create and save another STOP drawing.	drawing 32: STOP
4	Animations can be watched with the media player.	

Memory Game

Look at the picture, memorize it, press the start button. Try to reproduce it!
There are four modes available: Random segments, Numbers, Letters, Digital clock.
This game is really fun to play as a collaborative challenge with a team!

Button	Functionality [OFF / ON]	Remark
Latching_1	Play mode option 1	
Latching_2	Play mode option 2	
Latching_3	Menu / Play	In menu the generated pattern is already shown

Menu

Play mode defines which image will be generated on the display

Play Mode	Latching_1	Latching_2	Generate and display pattern
Random	OFF	OFF	Random segments (difficult mode)
Numbers	ON	OFF	Random digits (0000 to 9999)
Letters	OFF	ON	Random letters (AAAA to ZZZZ)
Clock	ON	ON	Random clock value (00.00 to 23.59)

Play

Step	Comment
1	Make sure Latching_3 is set to OFF
2	Memorize what’s displayed on the screen
3	Switch Latching_3 to ON
4	Try to reproduce the screen as it was shown at the start. See the draw app on how to draw on the display
5	When done with drawing, switch Latching_3 to OFF
6	If done correctly, a victory song will play. If you didn’t manage to reproduce the original drawing, the differences will be displayed.

Random world

Show various random generated events.

Button	Functionality [OFF / ON]	Remark
Latching_1	Shift	Each of the momentary buttons gets another function if ON.
Latching_2	animate?	"Rolling animation" if ON.
Latching_3	Manual / Auto draw	Auto draw: ● Set the draw interval time in seconds with the dial. ● Press a momentary button to start the auto draw Manual draw: ● Press a momentary button for each draw
Dial		Manual draw mode: ● rotate to draw

Random modes

Button	Shift ?	Function	
Momentary_0	No	Single Dice	Display shows eyes like a real dice

Momentary_0	Yes	Roll 4 dice at once	Each dice has its own digit.
			`display: VVVV`
			`V = [1-6]`

Momentary_1	No	Random Letter
		Shows random letter and its position in the alphabet
			`display: VV L`
			`VV = position of letter in alphabet`
			`L = representation of letter.`

Momentary_1	Yes	Draw the next card from a shuffled deck.
		The card is discarded at next draw. When all cards from the deck are gone, a new deck is shuffled.
		Sound will sound when a new deck is taken and shuffled.
			`display: V S`
			`V = value`
			`S = Suit`
			Values:
			1 = ACE,
			11 = JACK,
			12 = QUEEN
			13 = KING
			
			Suits:
			S = Spades
			C = Clubs
			H = Hearts
			D = Diamonds

Momentary_2	No	Random number in manual mode : hold button (minimum 2	

		seconds, and keep holding) to set upper limit with Dial.	
Momentary_2	Yes	Random sequence (=tombola) Custom Value [0-xxx] Set maximum value xxx by holding the button and turning the Dial. xxx will be saved in eeprom. in manual mode : hold button (minimum 2 seconds, and keep holding) to set upper limit with Dial. Will draw all numbers from the set (max 100 numbers) before starting a new set. Sound will sound when new sequence produced.	

Momentary_3	No	Coin toss	`display:`
			`HEAD (=head) or`
			`TAIL (=tails)`

Momentary_3	Yes	Psychic experience.
		Ask a question, your grand-grand aunty Lucy will respond with yes or no.
			`display:`
			`YES or`
			`NO`

Stopwatch

Two independent classic chronometers.

Control	**Functionality [OFF / ON]**	**Remark**
Latching_1	None	The rarest of beasts! A button without a function! Contact Uncle Lode if you have an idea.
Latching_2	Display maximum precision or seconds precision	
Latching_3	Stopwatch 1 / Stopwatch 2	The two stopwatch controls are not influencing each other
Momentary_0	Hold to show split time	
Momentary_1	Hold to save split time and show	
Momentary_2	Reset	
Momentary_3	Toggle Start Pause	

Geiger

A geiger counter detects radioactive decay. Every time radiation is sensed, a beep is emitted. The more beeps you hear, the more contaminated the environment is!

Note: There is no real radioactivity meter installed in the Luciebox. It’s all fake fake fake.

Button	**Functionality [OFF / ON]**	**Remark**
Latching_1	Geiger mode / Noise mode	
Latching_2	Animated / Counter	
Dial	sensitivity	

Geiger Mode

Button	**Function**	**Comment**
Momentary_0	None	Contact uncle Lode if you have a functionality idea
Momentary_1	None	Contact uncle Lode if you have a functionality idea
Momentary_2	None	Contact uncle Lode if you have a functionality idea
Momentary_3	Temporary sharp increase of tick probability	When pressed, it looks like there is a lot more radiation. **Practical joke idea**: Hold the box close to a person and press the button…. it then appears like the person is contaminated. Hilarity ensues! Warning, do not use this prank near microwave ovens. People are already paranoid enough about them as it is.
Dial	Set tick probability	

Noise Mode

Offers the possibility to tune Geiger as a random note generator.

Button	**Functionality**	**Comment**
Latching_3	Prolonged beep	Every beep will sound until another beep takes over if ON.
Momentary_0	Lower frequency	Keep pressed while rotating Dial(CW=higher, CCW = lower)
Momentary_1	Upper frequency	Keep pressed while rotating Dial (CW=higher, CCW = lower)
Momentary_2	Note length	Keep pressed while rotating Dial(CW=longer, CCW = shorter)
Momentary_3	trigger manually	Keep pressed while rotating dial for massive hits

Whack a mole, reaction game

Menu

Button	Functionality [OFF / ON]	Remark
Latching_1	Whack a mole / whack a bird	Can only be changed in Menu
Latching_2	Normal mode / Endurance mode	Can only be changed in Menu
Latching_3	Menu / Play	
Dial	Set level	Can only be changed in Menu

Play Whack a mole

Every momentary button represents a mole that's popping up. Your task is to hit it when it stick its head out! When the light goes on, you have to press the button as quick as possible to gain high scores.

Step	Function
1	In the menu, the level (L x) and its high score are shown intermittently. Choose a level with the dial.
2	Switch Latching_3 to ON to start playing
3	If a momentary button lights up (or a decimal point on the display), the corresponding button needs to be pressed.
4	When the button is pressed, the light goes out. If it stays on, the button needs to be pressed again as that persistent mole needs more whacking!
5	If a wrong button is pressed, or if time ran out, the game ends
6	At game end, your mole whacking score will be displayed. If you have a new high score, a victory song will be played.
7	After some seconds, a new game will start automatically. Go to step 3.
8	Switch Latching_3 to OFF to end the game and enter the menu at any time

Play Whack a bird

Step	Function
1	When the application is selected, four random notes from the chromatic scale are chosen. These notes stay the same for as long as the Whack-a-mole application is active.
2	In the menu, the level (L x) and its high score are shown intermittently. Choose a level with the dial.

3	Switch Latching_3 to ON to start playing
4	The four notes will be heard as a sequence. For each note, a button is assigned. It will light up when the sequence is playing.
5	Now a note is played. Press the corresponding momentary button to whack that bird. (In level one, rookie mode, the corresponding button is lit up as a visual clue.)
6	If a wrong button is pressed, or if time ran out, the game ends
7	At game end, your bird whacking score will be displayed. If you have a new high score, a victory song will be played.
8	After some seconds, a new game will start automatically. Go to step 4.
9	Switch Latching_3 to OFF to end the game and enter the menu at any time

Normal mode

The time assigned is the maximum time between two whacks.

Level	Maximum reaction time[s]	Comments
1	30	This seems long. But, run around the table or house for every mole you whack. How long can you last? What else can you do?
2	5	
3	2.5	
4	1	
5	~0.5	
6	~0.3	Obviously uncle Lode maxed out on this one!

Endurance mode

The time assigned is the total whacking time. Try to whack as many animals in that given time frame. Beware, one mistake and you have to start all over again…

level	Total whack time
1	2 minutes
2	1 minute
3	30 seconds
4	20 seconds
5	10 seconds
6	5 seconds

Adolescent

Simon

Simon says is a game where people try to repeat progressively longer sequences of button presses.

Main menu

Button	**Functionality [OFF / ON]**	
Latching_1	Simon / without Simon	Normally Simon decides the sequence. In games without Simon, every level, a player adds a step to the sequence.
Latching_2	All players repeat/ one player repeats	In multiplayer: all players have to enter the sequencer (computer decides player sequence) or only one random chosen player has to repeat it.
Latching_3	Main menu / Play	When playing, set to OFF to enable menu and reset game
Dial	set number of players	

Game selection

Latching_1	**Latching_2**	**Functionality implemented**	**desired functionality (future)**
0	0	Play with Simon. Every turn, the computer makes the sequence one step longer. In multiplayer, all players in random order (the number is displayed on the screen) have to repeat the sequence	
0	1	Play with Simon. Every turn, the computer makes the sequence one step longer. In multiplayer, only one player is chosen to repeat the sequence. The chosen player id is displayed.	
1	0	Play without Simon. Players build the sequence. The first player in a new rounds sets the new last step in the sequence. Every player repeats the sequence.	
1	1	Play without Simon. Players build the sequence. Only one random player has to repeat the sequence. He decides what the next last step is.	Pascal sake drinking game: Player presses a sequence (length decided by computer?) computer chooses another player who has to repeat it.

			TODO: Mode Pascal Sake Drinking Game **Use random switch AND custom build up options at the same time. For every round, a random player has to add a button press to the sequence.** **→ a lot more fun when not everybody has to do the new sequence.**

Play with Simon

Display	**Function**	**Comments**
`Sxx`	De computer player Simon shows a light sequence.	x = length of lights sequence [1-99]
`nPxx`	Player n has to repeat the light sequence Simon showed. Every button stands for a light.	Lights from left to right correspond with momentary buttons from left to right. n = player id [1 to 9]
`nPxx/ END (blink)`	All players are out. Last surviving player is n, with score: xx	Display blinks, during a repetition of the sequence one more time. After which a new game starts.

Play without Simon

Display	**Function**	**Comments**
`nPxx`	Player n has to repeat the light sequence that’s already set. Every button stands for a light. At the end of the sequence, the player adds one more step by pressing one of the buttons of his choice.	Lights from left to right correspond with momentary buttons from left to right. n = player id [1 to 9]
`nPxx/ END (blink)`	All players are out. Last surviving player is n, with score: xx	Display blinks, during a repetition of the sequence one more time. After which a new game starts.

Idea for the Pascal sake drinking game

Display	**Function**	**Comments**

nPxx	Player n inputs a certain sequence with the momentary buttons. The length of the sequence is defined by x	x = length of sequence [1-99]
		Use momentary buttons to input a sequence.
		n = player id [1 to 9]

nPpP	Player n has to repeat the light sequence Player p showed. This could be the same player (p could = n)	Use momentary buttons to input a sequence. n = player id [1 to 9]
pP / OUT (blink)	If player p makes a mistake, he’s out. Not sure if we continue then with a new sequence or if the next player has to input the correct sequence (this would ensure everybody keeps paying attention!)	Display blinks for 2 seconds, before continuing
nP / END (blink)	All players are out. Last surviving player is n	Display blinks for two seconds before restarting a new game

Guitar Hero

Button	**Functionality [OFF / ON]**	**Remark**
Latching_1	Guitar hero / Hex hero	
Latching_2	Normal / Extra mode	Extra mode: Guitar hero: Adds random paused Hex Hero: Input the complement of what you see for all digits with a dot. e.a. 6. → 1001
Latching_3	Menu / Play	
Dial	In Menu: Choose level	Higher level number = faster

Play Guitar Hero

Step	**Function**
1	In the menu, the level (L x) and its high score are shown intermittently. Choose a level with the dial.
2	Switch Latching_3 to ON to start playing
3	Lines fall down from the top of the screen.
4	Once lines have reached the bottom, you have to delete them by pressing the corresponding momentary button. Only delete lines that are AT the bottom.
5	The game ends if a button is pressed at the wrong time, or lines are not deleted in time.
6	At game end, your score will be displayed. If you have a new high score, a victory song

	will be played.
7	After some seconds, a new game will start automatically. Go to step 4.
8	Switch Latching_3 to OFF to end the game and enter the menu at any time

Team tip: Play guitar hero together: Every player holds his finger on one or more buttons.

Play Hex Hero

By the year 2036, having a reflexive knowledge of hexadecimal patterns will vastly improve your chances on the job market. Hex hero is an industry leading hex patterns trainer.

Step	Function
1	In the menu, the level (L x) and its high score are shown intermittently. Choose a level with the dial.
2	Switch Latching_3 to ON to start playing
3	Hexadecimal numbers characters scroll into the screen from the left
4	Press the corresponding nibble (=half byte) pattern with the momentary buttons to delete the most right hex number. Try to keep a clear screen. See the “Hex character to button pattern conversion table” below. In extra mode (Latching_2 ON), digits with a dot will appear. These need to have the inverted nibble pattern inputted. That’s called “the complement”.
5	The game ends if the hex numbers scroll out of the screen before they are deleted.
6	At game end, your score will be displayed. If you have a new high score, a victory song will be played.
7	After some seconds, a new game will start automatically. Go to step 4.
8	Switch Latching_3 to OFF to end the game and enter the menu at any time

Hex character to button pattern conversion table

Example: B shows up on the screen. B represents the value of 11 in decimal. In binary, B is 1011. Mimic this pattern with the momentary buttons by simultaneously pressing buttons 0, 2 and 3.

Character displayed	Decimal value	Converted to binary	Required button pattern to be pressed simultaneously				Extra: Complement
			Momentary_0	Momentary_1	Momentary_2	Momentary_3	
1	1	0001				x	1110
2	2	0010			x		1101

3	3	0011			x	x	1100
4	4	0100		x			1011
5	5	0101		x		x	1010
6	6	0110		x	x		1001
7	7	0111		x	x	x	1000
8	8	1000	x				0111
9	9	1001	x			x	0110
A	10	1010	x		x		0101
B	11	1011	x		x	x	0100
C	12	1100	x	x			0011
D	13	1101	x	x		x	0010
E	14	1110	x	x	x		0001
F	15	1111	x	x	x	x	0000

Quiz master

~~One person is the quiz master. Up to four players are the participants. When the quiz master presses the button, a random timer is started. After that time is elapsed, the lights of the player buttons go on. Every player has a button. First one who presses can answer the question. If the player is correct, he keeps his button pressed while the quizmaster resets his button. That way, a point is added to his score. And off to the next round!~~

Button	**Functionality [OFF / ON]**	**Remark**
Latching_1	Normal / Long maximum wait time	● normal: max 3 seconds ● long: max 20 seconds (which is very long!)
Latching_2	Quizmode / Independent mode	in independent mode, this is a reaction game. At the random intervals, the light goes on. First player pressing gets a point. auto play some rounds.
Latching_3	Question asked	Quizmaster “just asked a question button” In independent mode: ON=play game
Dial		

Quizmaster

Step	**Comments**

1	Every player gets a momentary button assigned.
2	Quizmaster asks question
3	Quizmaster sets Latching_3 to ON
4	When the momentary buttons light up, the players can press their corresponding button. The first players pressing its button can answer. A player that presses its button before it’s lit up will have its score reset.
5	Quizmaster can now turn dial to add or subtract a point to the player’s score
6	Quizmaster sets Latching_3 to OFF
7	Go to step 2

Independent reaction game mode

In independent mode, this becomes a very fun multiplayer reaction game.

Step	**Comments**
1	Every player gets a momentary button assigned.
2	Start the game by setting Latching_3 to ON
3	Wait until the button lights go on.
4	The first player to press its corresponding button (when the light is on) gets a point. All players that press their buttons before it’s lit up will have their score reset.
5	First player to reach ten points will hear a victory song.
6	Reset the game with Latching_3

Movie Player

Combine animations with songs to create full featured movies.

Functionality	**Latching Button**				**Momentary Button**				**Dial**
	0	**1**	**2**	**3**	**0**	**1**	**2**	**3**	
Sound Off (=Mute button)	1		1						
Sound On	1		0						
Autoplay movies	1			0					
Set playback speed	1	0		0					Turn
Manual movie mode	1			1					

Scroll through frames	1	0		1					Turn
Play song	1	1			1	0	0	0	
Toggle between continuous repeat of song and only play at start of movie	1	1			0	1	0	0	
Previous Song	1	1			0	0	1	0	
Next Song	1	1			0	0	0	1	
Next Movie	1	0			1	0	0	0	
Display Inverted	1	0			0	1	0	0	
Previous frame	1	0		1	0	0	1	0	
Next frame	1	0		1	0	0	0	1	
Play movie forwards	1	0		0	0	0	1	0	
Play movie backwards	1	0		0	0	0	0	1	

Animation selection

Cycle through all the animations with the “next movie” button.
Sequence: built in animations - home made animations
Note: See DRAW app on how to create a home made animation

Song selection

Cycle through all the songs with the next and previous buttons.
Sequence: 8 built in songs - 4 composed songs - 4 sequencer songs

Pomodoro Performance Meter

The Pomodoro Timer is a well known tool to be able to concentrate for specific task such as studying, filing in taxes, doing homework, cleaning, …
https://en.wikipedia.org/wiki/Pomodoro_Technique

The implementation tries to imitate the original as closely as possible. It also has an experimental performance meter. When enabled, the timer will beep at random intervals. At these beeps, you have to press the “Yes, I was working button” or the “no, I was procrastination or dreaming” button. A tally is kept of the key presses, so you can see your good vs bad ratio. This is the computer equivalent of your mother checking up on you at unexpected moments!

The official Pomodoro timer is a kitchen timer which has to be wind up manually. This very physical act prepares the mind for the task at hand. But, as we are automation aficionado's, an auto mode was added that automatically restarts the cycle after a settable pause.

When enabled, during the Pomodoro timing, at random intervals, beeps will be heard. When a beep is heard, a button needs to be pressed. Assign your values to the buttons. Good, bad, sleeping….. The buttons can be pressed at any time. This way, they can serve also as a manual counter. e.a.how many sit-ups can you do in two minutes? Four kids playing, one person pressing buttons per sit-up. Or kids running back and forth from the wall to the timer for one minute, and pressing their button every time they pass… Yeah, I'm up to something

Menu

Control	Functionality [OFF / ON]	Remark
Latching_1	Enable Break time and auto restart	
Latching_2	Enable Visual Timer	Visual timer shows the amount of segments lit up as a function of the left over time. This is easier to grasp for young children.
Latching_3	Switch ON to start timer	Once started the settings will be saved to the permanent eeprom memory, to be retrieved later.
dial	if no momentary_0 or momentary_1 is pressed, will set the set POMODORO time	The official Pomodoro time is 25 minutes.
Momentary_0 + dial	Hold and turn Dial to set POMODORO break duration.	The official Pomodoro break time is 5 minutes.
Momentary_1 + dial	Hold and turn Dial to set random beep probability interval.	Must be bigger than zero to enable. Set the probability time at which interval a beep will be heard. This means that "on average" at this interval a beep will be heard. The randomization is to ensure that we are being checked up on at irregular intervals.

Timing

Control	Functionality [OFF / ON]	Remark
Latching_1	enable/ disable break	
Latching_2	Display Visual timer.	
Latching_3	If released, the timer will stop immediately.	
Dial	Set volume and number of ticks.	The official Pomodoro guide states that the audible ticks remind us of the task at hand and help us to stay

		focussed.
Momentary_0	See number of NO’s	
Momentary_1	See number of Yes’s	
Momentary_2	add 1 to NO	
Momentary_3	add 1 to YES	

MultiTimer

A chess clock for up to four players.

Button	**Functionality [OFF / ON]**	**Comments**
Latching_3	Menu / Timing	Switching to OFF will always reset the timer

Menu

Button	**Functionality [OFF / ON]**	**Comments**
Latching_1	Fischer Timer	● Set the time that will be added to every timer after its turn is finished. ● The Fischer timer prevents a player from not using all its time at once. It provides a fighting chance to come back into the game if a player is almost out of time. ● As copied from boardgamearena.com: When playing a timer’s time will never exceed its initial time.
Latching_2	Random	● A random timer will be chosen at game startup. ● When selected, all enabled timer button lights are blinking to indicate a random starter.
Latching_3	Menu / Start Timing	
Dial	Set number of timers, Set time	● Normal: Set the number of timers. ● If corresponding momentary buttons is pushed: Set a timer’s initial time
Momentary_buttons		● If the light is on, the player is selected to join the game. ● If it is blinking, it is the starting player. ● To set the time of the player, press the button while rotating the dial.

Timing mode

Button	**Functionality [OFF / ON]**	**Comments**

Latching_1	None	
Latching_2	Pause / Unpause	
Latching_3	Menu / Start Timing	
Momentary _buttons	Next timer	● If blinking slow: Timer is selected and ticking. Press the button to select the next timer ● If solid: Timer is not selected and not elapsed ● If fast blinking in pause mode: Selected but paused timer ● If fast blinking not in pause mode: checking remaining time while timer not selected ● If Off: Timer elapsed or not initialized

Professional

Sequencer

A sequencer indefinitely plays a programmed pattern.
The sequencer has a total length of 32 steps spread out over four bars of 8 steps. The bottom four horizontal segments on the display indicate the active bar, the top and middle horizontal segments indicate the active step on that bar. Each step can contain one note.
At startup of the app a random sequencer song will be generated for you. It will change at every restart of the app. If you like it, you'll have to store it manually.

Button	**Functionality [OFF / ON]**	**Comments**
Latching_1	Normal mode / Eeprom Menu	

Normal Mode

Button	**Functionality [OFF / ON]**	**Comments**
Latching_2	one bar / all bars	If ON: When assigning a note to a step, it will be assigned to the step of all 4 bars. VERY HANDY. Use this to create a “base” melody that is the same for all 4 bars.
Latching_3	Manual / Autoplay	If no momentary buttons pressed, dial will set speed in autoplay mode
Momentary_0 Manual mode	Play and show assigned note at active step	
Momentary_0	Hold to show	BONUS: hold and rotate dial to transpose the

Autoplay	assigned notes at active step as they are played	programmed sequence in auto play mode
Momentary_1	Set and listen to active note without assigning it	hold and rotate dial to choose the active note
Momentary_2	Assign active note to active slot	rotate the dial to choose an active note (will not be shown on display)
Momentary_3	Next step	Hold down and rotate dial to quickly go forward or backward

Eeprom menu

Button	Functionality [OFF / ON]	Comments
Latching_3	LOAD / SAVE	Should a song be saved or loaded?
Momentary_1	Execute	Will save to eeprom or load from eeprom (depending on latching_3)
Dial	Choose song number	

Song composer

Create your own songs and store them permanently in eeprom memory. Or load and edit a song. Each song has a maximum length of 100 notes.

Button	Functionality [OFF / ON]	Comments
Latching_1	Compose mode / Eeprom menu	Compose a song or save or load a song from Eeprom
Latching_2	Note mode / Index mode	Display note value / Display note position in song
Dial	in autoplay: set speed, in manual mode: scroll through all notes.	

Compose mode

Button	Functionality [OFF / ON]	
Momentary_0 in Note mode	Play active note	
Momentary_1 in Note mode	Select note	Hold down and turn dial to select active note from scale

Press Momentary_0 while holding Momentary_1 in Note mode	Program note	It’s important to FIRST hold and keep holding Momentary_1, and then, to program, shortly press Momenary_0
Momentary_0 in Index Mode	Delete current note in song	Delete slot. Move all notes so there is no gap
Momentary_1 in Index Mode	Insert a note in the song	Insert slot behind current location. The song gets longer. If longer than the maximum allowed number of notes, the last note is erased.
Momentary_2	Previous note in song	
Momentary_3	Next note in song	
Latching_3	Manual / Auto	If ON: the note sequence is played automatically
Dial	● Auto play: set speed ● Manual play: scroll through song.. ● With momentary_1 in note mode: select note	

Eeprom menu

Button	**Functionality [OFF / ON]**	
Latching_3	SAVE / LOAD	Should a song be saved or loaded
Momentary_0	Execute	Save to eeprom or load from eeprom (depending on latching_3)
Dial	Choose program number	

Uncle Lode’s crazy level

Hacktime

See the FLASH, RAM and EEPROM memories in their rawest state!
Scroll manually or automatically through all the bytes. Let them make noise or just display them on the screen. Show the index or the value of the address. Modify the RAM and EEPROM manually. Can you cheat your way into the game highscores? If things go wrong, just switch the box on and off again.

note: Active address is the most left digit on the display

Modes

Button	Function [OFF / ON]	Remark
Latching_1	Address mode / normal mode	
Latching_2	Mute button	Sound OFF / ON. Every byte encodes for a note. On every change, this note is played. Be prepared for post-modernist masterpieces in auto-mode.
Latching_3	Auto scroll / Manual	
Momentary_2	HOLD and rotate Dial to move memory location x255	
Momentary_3	HOLD and rotate Dial to move memory location x4	4 is chosen, because the display can hold four bytes at its most.
Dial		● Auto scroll: Change speed ● Manual: Change address

Address mode

Control	Functionality	Screen
Momentary_0	Set memory type	Cycle through: ● FLASH: displays F ● RAM: displays R ● EEPROM: displays E

Normal mode

Control	Functionality	Screen
Momentary_1 with dial		Cycle through: ● values displayed as ASCII-like character ● values displayed as bytes (7 segments + decimal point = 8 bits represented) ● most left digit on display: decimal ● most left digit on display: hexadecimal
Dial	while Momentary_1 is pressed, write	Hold and rotate Dial to change the active memory byte value from 0 to 255. ~~When the value changes in the memory, a confirmation beep will sound~~. Will work for eeprom and ram, but not for flash. Beware: when changing RAM, it might change back to another value right away as you're tweaking the cpu live. Also, the device can suddenly behave strangely. Alternatively, the universe might collapse.

Value representation

Decimal representation

Value from 0 to 255 (left character on the display)

Hexadecimal representation

Value from 0 to 0xFF (left segment on the display)

Binary representation

7 segment Digit, segment to byte position. light on = 1

```
    0
 5      1
    6
 4      2
    3        7


           XXXX
              X
           XXXX
           X  X
e.a.       X  X = h57 = b01010111
```

Segment name	Byte bit index	Segment Position in Digit description
A	0	top
B	1	up right
C	2	bottom right
D	3	bottom
E	4	bottom left
F	5	top left
G	6	center
DP	7	decimal point

Character representation

All values that are not indicated in the table will have their value represented as a blank (just like a space)

Character	Binary to 7 segment (left	ASCII value	Remark

	is A, right is DP)		
‘ ‘			Space is not implemented, provide SPACE_FAKE_ASCII. It will still work, because all unrecognized code will resolved into spaces.
0	B00111111	48	
1	B00000110	49	
2	B01011011	50	
3	B01001111	51	
4	B01100110	52	
5	B01101101	53	
6	B01111101	54	
7	B00000111	55	
8	B01111111	56	
9	B01101111	57	
ONLY_TOP_SEGMENT_FAKE_ASCII	B00000001	58	Not an official ASCII character
ONLY_MIDDLE_SEGMENT_FAKE_ASCII	B01000000	59	Not an official ASCII character
ONLY_BOTTOM_SEGMENT_FAKE_ASCII	B00001000	60	Not an official ASCII character
ONLY_TOP_AND_BOTTOM_SEGMENT_FAKE_ASCII	B00001001	61	Not an official ASCII character
SPACE_FAKE_ASCII	B00000000	62	Not an official ASCII character
RANDOM	RANDOM	63	Not an official ASCII character, SPECIAL CASE, will randomly toggle one of the segment each time when assigned.
	B00000000	64	
A	B01110111	65	
B	B01111100	66	
C	B00111001	67	

D	B01011110	68	
E	B01111001	69	
F	B01110001	70	
G	B00111101	71	
H	B01110100	72	
I	B00000110	73	
J	B00001110	74	
K	B01110101	75	
L	B00111000	76	
M	B01010101	77	
N	B01010100	78	
O	B01011100	79	
P	B01110011	80	
Q	B01100111	81	
R	B01010000	82	
S	B01101101	83	
T	B01111000	84	
U	B00011100	85	
V	B00011110	86	
W	B00011101	87	
X	B00110110	88	
Y	B01101110	89	
Z	B00011011	90	

Memory types

Memory Type	**Address range Decimal**	**Address range Hexadecimal**	**Length [bytes]**	**Editable**	**Retains value when powered off**	**Function**
Flash	0..32255	0..7DFF	32255	NO	YES	This memory contains the main program code. It is

						programmable to upload the firmware, but it cannot be changed from within the program.
Ram	0..2047	0..7FF	2048	YES	NO	This fast memory is needed for the program to function
Eeprom	0..1023	0..3FF	1024	YES	YES	This slow memory contains values the need to be preserved

Luciebox eeprom layout

The flash and ram memory are organised beyond my control. But, the eeprom has 1024 bytes that can be programmed and will retain their value, even when the box is switched on and off again. It is strictly organised according to the schematic below.

Address range Decimal	Address range Hexadecimal	Length [bytes]	function	
0..7	0..7	8	Settings	● byte 0: Sound disabled ● byte 1..2: Power cycle counter (will only update if at least one of the momentary buttons is pressed after a switch on)
8.. 103	8..67	96	Game high scores	each game has 6 levels. High score is 2 bytes (because I choose to believe in your abilities!) ● byte 0..11: Whack a mole ● byte 12..23: Whack a mole endurance mode ● byte 24..35: Whack a bird ● byte 36..47: Whack a bird endurance mode ● byte 48..59: Guitar Hero ● byte 60..71: Guitar Hero with stuttering ● byte 72..83: Hex Hero ● byte 84..95: Hex Hero with stuttering
104..109	68..6D	6	MultiTimer	● byte 0: Number of enabled timers ● byte 1..4: Init timer per timer [seconds] ● byte 5: Fischer time [seconds]
110..112	6E..70	3	Pomodoro	● byte 0: Init time [seconds] ● byte 1: Pause time [seconds] ● byte 2: Random beep time [seconds]
113..240	71..F0	128	Sequencer	32 bytes per sequence * 4 sequences
241..640	F1..280	400	Composer songs	100 bytes per song * 4 songs
641..1020	281..3FC	380	Sketch	4 bytes per picture* 95 pictures

			saved Pictures	
1021..1023	3FD..3FF	3	Free	Free space. All for you to store secret information, passwords, bitcoin passwords,...

Practical

Inactivity timer

Every hour since the last key press, the happy dryer song will play. Unless the Pomodoro Application or Multitimer is activated.

Randomness

https://rheingoldheavy.com/better-arduino-random-values/
The random seed is done with the internal timer at each app initialization. So, at startup, as long as not app has been changed, the same values will show up for generated random event (rolling dice, sequencer generator, guitar hero sequence,...). It's important that your sequence does not involve button presses, because, these cannot be timed. There is something fun about the idea of have the same randomness. For better randomness, make sure to change an app some time after startup of the box.

Settings App

Button test and Luciebox settings.
Decimal points on display flicker erratically to indicate that the box is alive.
This is a simple buttons test app, that is very appealing to small kids as the behaviour is very predictable. All button presses have an instant effect!

Latching Button				**Momentary Button**				**Dial**	**Functionality**
0	**1**	**2**	**3**	**0**	**1**	**2**	**3**		
1	1	1	1						Corresponding digit displays "-"
				1	1	1	1		Corresponding digit displays "0"
								turn	Twist back and forth 3 times to activate the settings menu. Every consecutive twist will activate the next menu item.

Settings Menu items

Dial twists	Menu item	Functionality
3	Enable sound	With momentary 0, toggle the sound. This will be stored in the

		retentive memory.
4	A0	The value of the selector dial. Stays at about 680. Cannot be changed.
5	A1	The value is an addition of pressed momentary buttons. Momentary_0:~64, momentary_1: ~128, momentary_2: ~256, momentary_3: ~512
6	A2	The value is an addition of pressed latching buttons. Latching_0:~64, Latching_1: ~128, Latching_2: ~256, Latching_3: ~512
7	Firmware version	Have this number ready when contacting the helpdesk.
8	QTY	Show the amount of times the Luciebox was switched on and at least one button was pressed.
9	Reset	Press M0 and then M3 for game high scores reset Press M0 and then M2 for total reset (includes drawings, composed songs, high scores, sequencer songs)

Troubleshooting

Problem	**Comment**
App sometimes restarts. Some apps show a blank screen. Some apps are missing.	The selector dial wiper might have a gap. Bad bad selector switch. With a small screwdriver, bend the wiper downward to eliminate the gap with the outer conducting circle.
Luciebox does not start up, or only for some seconds	Undervoltage, batteries empty.
I have no clue what to do.	Read this manual.
I can’t read yet	Vraag hulp aan een ouder of voogd.
Ik ken geen Engels	Vraag maar aan Omi, zij heeft Engelse les gevolgd!

Todo and issues

F302 Changes

Issue	Status
metronome:button3 hold: set sound also when metronome ON. Change flashing toggle to another button.	No don’t do this. ok like it is. close issue
check if the MAX from random number (bingo) is indeed saved to eeprom once set…	Done
in random mode: dial function: not working. : If latching_2 and latching_3 are off: Generate random values for last selected mode. (dial should be like next step)	done. Does it add value though?
Mode draw letters: would be fun to have the probability of drawn letter set to a certain language. English, Dutch,.... this way it can be used in word games.	done in flash Hmmm… not enough memory for such a multi language endeavour for now. I could work if we put the probabilities per letter as byte in the eeprom. So, it can be changed in hack mode. Then, to pick a letter: make sum of all probabilities. Choose random number between zero and that total. In a for loop, add the probabilities again, if the chosen random number, is in an interval, that’s our number! aka: previous probability index when counting up. Of, just store the english letter frequencies in the ram. It’s not thaaaat different from dutch...
chronometer (or countdown timer?) Set animation, instead of numbers on screen, show hourglass on screen for countdown	we could use the screen fill animation. **not done , not needed close**
simon says multiplayer mode: do not let everybody do the sequence. Just one person. let random mode decide if it’s just the next person, or a random person	OK done.
check pascal “No Simon” mode.	OK done
todo quizmaster app: dial sets score in quizmaster mode after question answered. → press all buttons to which a value is added or subtracted. hmm easier: dial changes value of player that pressed	ok done
multitimer: multiple timers, next player: not working!	ugh, no problem. works! did you read the manual on how to use it ? Closed.
enable random pause in hex hero	that’s silly. But, what we could do: complementary mode: whenever the decimal point is ON, the complement of the given number has to be provided done
song composer: check if dial rotating with If momentary 1	reworked done

or 2 is pressed, sets note.	
song composer: hold down momentary0 and rotate dial will not assign notes while dialing	reworked done
hack time: When the value changes in the memory, a confirmation beep will sound → NOPE	solved.
hacktime: representation of values as chars and binary looks like it's not correct	done
inactivity timer	done beep every X minutes.(30?) → one hour

Remarks

Power solutions

- Batteries: 4xAA → this is more like 6.5V when the batteries are new, but, the Luciebox can handle (thanks to a Schottky diode)
- USB Mini connected to a powerbank or a phone charger

Firmware update

Check the firmware version in the settings app. An update or your own software can be uploaded.

Background

The arduino platform is used. The arduino bootloader is loaded in the Microcontroller. Imagine the Luciebox as an Arduino UNO board without its microcontroller. When the wires are connected like in the table, programming the Luciebox is like uploading a program to any Arduino Uno board. Experimenting is encouraged.

Upload firmware procedure

1. Have an arduino UNO board ready. These boards are available for very cheap on aliexpress.
2. Take an arduino uno board, and connect the wires like indicated in the table below. There is no need to modify the arduino uno board. (For serial output to work in debug mode (e.a. Serial.println()), the atmega328 needs to be removed from the original UNO).
3. Open the arduino IDE and load the Luciebox program.
4. Upload the program to the connected "arduino UNO".
5. Disconnect the RX/ TX pins if the atmega328 is not removed from the UNO board.
6. Done. (you can leave the VCC and GND connected to save on batteries).

Luciebox to Arduino UNO interface

Pin Arduino programmer	Pin Luciebox	Remarks
VCC	VCC	5V DC
GND	GND	Ground
TX	TX	Luciebox to Computer
RX	RX	Computer to Luciebox
RESET	RESET	Pulled to GND during programming.