# Luciebox

## Owners manual

The Luciebox pretbak, made by Uncle Lode Ameije for Lucie.
Project started 2019-04-02. Manual v20211005

# 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! I realize now there is no such thing as finished projects, only abandoned projects. I'm welcoming all suggestions and contributions, write them down as comments in the appropriate section in this document. Looking at you Lucie! Regards from your high-expectations-uncle Lode.

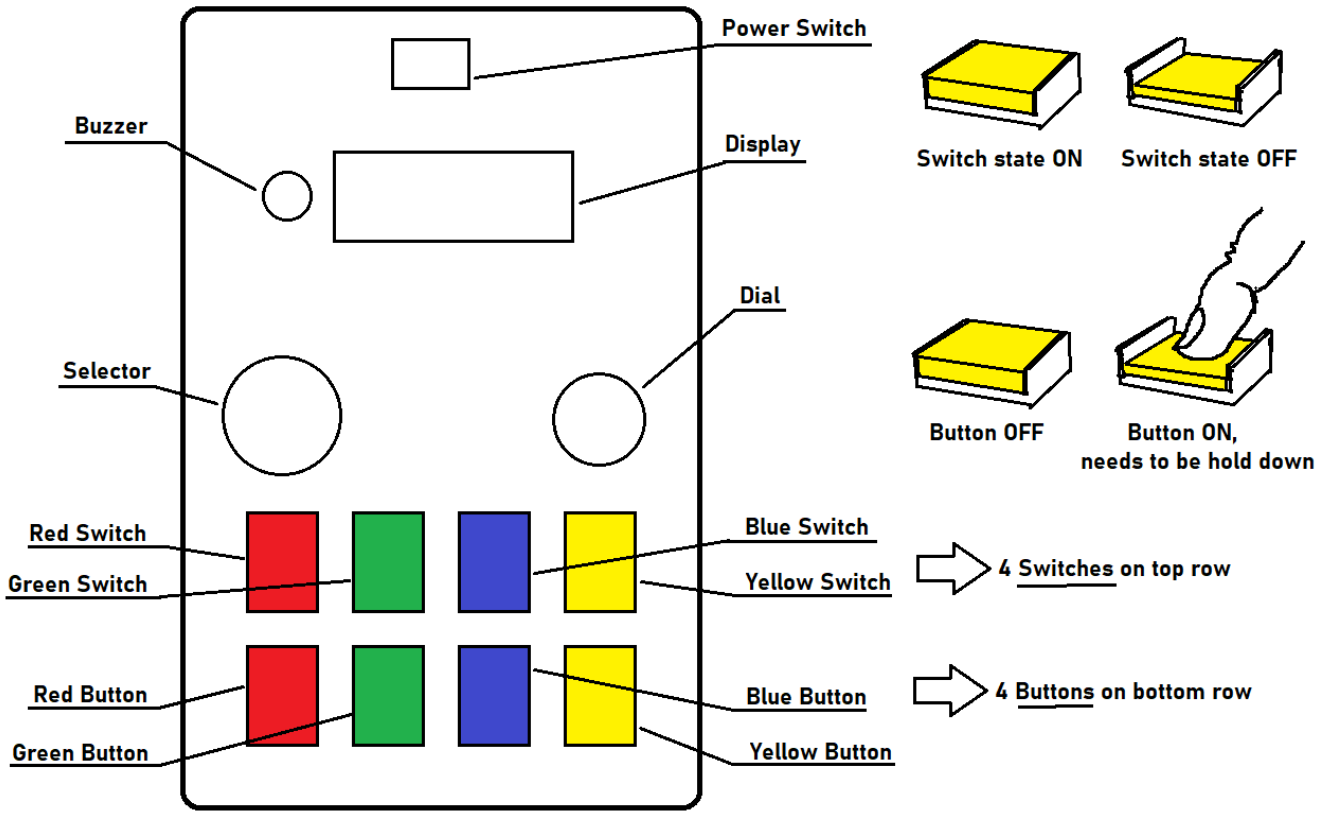
# How to use this box

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 the start leave all 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.

| **Application** | **Selector Position** | **Red Switch Position** | **Splash screen** | **Comments** | **Difficulty [1-5]** |
|---|---|---|---|---|---|
| Test and Settings | 1 | 0 |  wrench | Test all the buttons. Check and modify the Luciebox configuration. | 1 |
| | 1 | 1 |  | | |
| Stopwatch | 2 | 0 |  stopwatch | All the goodies from a typical 80's stopwatch. Has two independent timers. | 2 |
| MultiTimer | | 1 |  | A game clock for up to 4 players. Can also be used as a chess clock or as a kitchen timer. | 2 |
| School | 3 | 0 |  koala / Adhemar | Learn how to count, learn the alphabet | 3 |
| Pomodoro Timer | 3 | 1 |  | The Pomodoro technique helps procrastinators to get things done | 3 |

| | | | | | |
|---|---|---|---|---|---|
| Random world | 4 | 0 |  dice | Roll the dice, flip a coin, play bingo, draw random playing cards or let psychic Lucie answer all your questions. | 2 |
| Score tracker | 4 | 1 |  | Keeps a tally of the score of up to four players. | 2 |
| Simon Says Game | 5 | 0 |  face | 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 |  gun | 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 |  pencil | Scroll through and edit 95 pictures | 3 |
| Memory game | 7 | 1 |  | Reproduce what the display showed. | 3 |
| Radio | 8 | 0 |  movie reel | Play the main songs, your own compositions and the stored sequencer loops | 1 |
| Movie player | 8 | 1 |  | A full featured media player. Combine the built in or stored animations with one of the songs. It could as well have been named "my first meme". | 3 |
| Metronome | 9 | 0 |  drumkit | 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 |  note | Experiment with tones, lengths, scales and arpeggios. Features an auto mode | 1 |

| | | | | | |
|---|---|---|---|---|---|
| Song Composer | 10 | 1 |  | Create and save up to four songs. | 4 |
| Noise Generator | 11 | 0 |  radioactive | Measure your local radio activity. Or generate all kinds of noise. | 1 |
| Hack Time | 11 | 1 |  | Display and edit the Luciebox memory. Experiment as you like. | 5 |
| Dream Time | 12 | 0 |  manta ray | Baby action with simple button actions, patterns and sounds. | 1 |
| | 12 | 1 |  | | 1 |

# Toddler

## Radio

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

| Input | Functionality | Comments |
|---|---|---|
| Green Switch | Song bank selector 1 | A combination of the bank selector assigns different songs to the momentary buttons |
| Blue Switch | Song bank selector 2 | A combination of the bank selector assigns different songs to the momentary buttons |
| Yellow Switch | Playback speed / Transpose | Dial will change the settings while a song is playing |

### Songs

| Green Switch | Blue Switch | 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.

| Input | Functionality [OFF / ON] | Comments |
|---|---|---|
| Green Switch | Piano mode / Advanced mode | |
| Blue Switch | 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. |
| Yellow Switch | Manual mode / Auto arpeggio | Auto arpeggio will automatically play the next note after a timed interval |
| Dial | | in auto arpeggio: set interval,<br>in manual mode: go up or down the scale |

## Piano mode

| Input | Functionality | Comments |
|---|---|---|
| Red Button | Mute | Will not make sound while changing notes. This allows you to jump notes silently when playing a song. |
| Green Button | Play active note | Play the last played note |
| Blue Button | Previous note | Previous note on the active scale |
| Yellow Button | Next note | Next note on the active scale |

## Advanced mode

| Input | Functionality | Comments |
|---|---|---|
| Red Button | Duration | Cycle between full, half, quarter, eight notes |
| Green Button | Progression | Cycle between up/down patterns:<br>● MANU: Manual (default, set direction yourself),<br>● UP: Going up the scale,<br>● DO: Going down the scale,<br>● UPDO: Saw pattern, going up and down, changing at a random interval<br>● RND: Random picking of a note on the scale<br>● CRAY: Crazy mode going randomly up and down at erratic intervals. |
| Blue Button | Scale | Cycle between different scales:<br>● AJOR: Major<br>● INOR: Minor<br>● PENT: Pentatonic<br>● BLUE: Blues Major<br>● CHRO: Chromatic |

| | | |
|---|---|---|
| Yellow Button | Key | Cycle between all 12 tones in an octave to set the key of the selected scale. Press once to display the current key. Next button presses will change it. |

# Metronome

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

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Green Switch | Make ticker 2 step backwards | |
| Blue Switch | Set speed in beats per minute (bpm) | One ‘beat’ is a full circle of 12 steps. |
| Yellow Switch | Manual mode / Ticking mode | |
| Red Button | Reset the positions of all tickers | Set to the position of ticker 1 |
| Green Button | Increase ticker 2 offset by one | |
| Blue Button | Increase ticker 3 offset by one | |

## Metronome Manual mode

| **Yellow Button** | **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.<br>Shorter beeps lose their pitch, and are more like a metronome ‘puck’ sound. |

## Metronome Ticking mode

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

# Dream Time

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

| Input | Functionality [OFF / ON] | Comments |
|---|---|---|
| Green Switch | Play song | Toggle between Alphabet song and Happy Dryer song |
| Blue Switch | 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. |
| Yellow Switch | Auto / Manual | Dial will change step in manual and set interval speed in auto mode |
| Red Button | Uncle Lode Indoctrination | |
| Green Button | PRRRRR noise | |
| Blue Button | Previous step | |
| Yellow Button | Next step | |

# Child

## Easy timer

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

## School

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.

| Input | Functionality [OFF / ON] | Comments |
|---|---|---|
| Green Switch | Number mode / alphabet mode | display numbers or letters |
| Blue Switch | Numbers as Decimal / Hexadecimal | Only effective in number mode |
| Yellow Switch | Manual / Auto count | Auto mode<br>● Song is played when in alphabet mode<br>● Dial changes the speed<br>Manual mode<br>● Dial changes value |
| Red Button | in auto mode: toggle count up / down | |
| Green Button | set value zero | |

| | | |
|---|---|---|
| Blue Button | value down | |
| Yellow Button | value up | |

## Score tracker

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

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Green Switch | Count up / Count down | |
| Blue Switch | Apply score change to single tally / to all tallys | |
| Yellow Switch | Modify mode / View mode | In view mode, every tally can be checked without modifying it |
| Red Button | Tally 1 | |
| Green Button | Tally 2 | |
| Blue Button | Tally 3 | |
| Yellow Button | 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.

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Green Switch | go through all four positions for a full cycle / skip lateral movements for a full cycle. | don’t take lateral movement into account |
| Blue Switch | | |
| Yellow Switch | | |
| 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

| **Input** | **Functionality [OFF / ON]** | **Remark** |
|---|---|---|
| Green Switch | ON: Save menu | Save a drawing to the eeprom memory |
| Blue Switch | Display Image / Display Index | “Display index” displays the eeprom drawing address |
| Yellow Switch | View / Draw | In draw mode, drawings can be edited |

## Display mode

| **Input** | **Functionality** | **Screen** |
|---|---|---|
| Red Button | | |
| Green Button | Global display setting | cycle through:<br>● normal (default)<br>● negative<br>● all on<br>● all off<br>● STOP screen |
| Blue Button | Previous drawing | |
| Yellow Button | Next drawing | |
| Dial | Scroll through drawings | |

## Draw mode

| **Input** | **Functionality** | **Screen** |
|---|---|---|
| Red Button | Toggle active segment | |
| Green Button | Global display setting | cycle through:<br>● normal (default)<br>● negative<br>● all on<br>● all off<br>● STOP screen |
| Blue Button | Move cursor inside digit | |
| Yellow Button | 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 .

| Input | Functionality | Comments |
|---|---|---|
| Red Button | Save screen | Will save the drawing to the active EEPROM address (visible with Green Switch) |
| Green Button | HOLD to have “shift function for momenary_2 and Yellow Button | |
| Blue Button | previous  drawing OR with SHIFT FUNCTION: delete drawing slot (and move all indexes, ) | Hold down Green Button for it to work |
| Yellow Button | 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 Green Button 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 | Go to the next drawing. Now make as many drawings as you want in the next slots. These are the frames of the movie. | drawing 28: UNCL<br>drawing 29: LODE<br>drawing 30: IS<br>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 “Movie Player” app. | |

## 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!

| Input | Functionality [OFF / ON] | Remark |
|---|---|---|
| Green Switch | Play mode option 1 | |

| | | |
|---|---|---|
| Blue Switch | Play mode option 2 | |
| Yellow Switch | Menu / Play | In menu the generated pattern is already shown |

### Menu

Play mode defines which image will be generated on the display

| **Play Mode** | **Green Switch** | **Blue Switch** | **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 Yellow Switch is set to OFF |
| 2 | Memorize what’s displayed on the screen |
| 3 | Switch Yellow Switch 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 Yellow Switch 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** |
|---|---|---|
| Green Switch | Shift | Each of the momentary buttons gets another function if ON. |
| Blue Switch | animate? | “Rolling animation” if ON. |

| | | |
|---|---|---|
| Yellow Switch | Manual / Auto draw | Auto draw:<br>● Set the draw interval time in seconds with the dial.<br>● Press a momentary button to start the auto draw<br>Manual draw:<br>● Press a momentary button for each draw |
| Dial | | Manual draw mode:<br>● rotate to draw |

## Random modes

| **Input** | **Shift?** | **Function** | |
|---|---|---|---|
| Red Button | No | Single Dice | Display shows eyes like a real dice |
| Red Button | Yes | Roll 4 dice at once | Each dice has its own digit.<br>`display: VVVV`<br>`V = [1-6]` |
| Green Button | No | Random Letter<br>Shows random letter and its position in the alphabet | `display: VV L`<br>`VV = position of letter in alphabet`<br>`L = representation of letter.` |
| Green Button | Yes | Draw the next card from a shuffled deck.<br>The card is discarded at next draw. When all cards from the deck are gone, a new deck is shuffled.<br>Sound will sound when a new deck is taken and shuffled. | `display: V S`<br>`V = value`<br>`S = Suit`<br>Values:<br>1 = ACE,<br>11 = JACK,<br>12 = QUEEN<br>13 = KING<br><br>Suits:<br>S = Spades<br>C = Clubs<br>H = Hearts<br>D = Diamonds |
| Blue Button | No | Random number<br>in manual mode : hold button (minimum 2 seconds, and keep holding) to set upper limit with Dial. | |
| Blue Button | Yes | Random sequence (=tombola)<br><br>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.<br>Will draw all numbers from the set (max 100 numbers) before starting a new set.<br>Sound will sound when new sequence produced. | |
| Yellow Button | No | Coin toss | `display:`<br>`HEAD (=head) or`<br>`TAIL (=tails)` |
| Yellow Button | Yes | Psychic experience.<br>Ask a question, your grand-grand aunty Lucy will respond with yes or no. | `display:`<br>`YES or`<br>`NO` |

## Stopwatch

Two independent classic chronometers.
At startup of the app, the second chronometer contains the total ON time since the Lucie box was switched on.

| **Input** | **Functionality [OFF / ON]** | **Remark** |
|---|---|---|
| Green Switch | None | The rarest of beasts! A button without a function! Contact Uncle Lode if you have an idea. |
| Blue Switch | Display maximum precision or seconds precision | |
| Yellow Switch | Stopwatch 1 / Stopwatch 2 | The two stopwatch controls are not influencing each other |
| Red Button | Hold to show split time | |
| Green Button | Hold to save split time and show | |
| Blue Button | Reset | |
| Yellow Button | Toggle Start Pause | |

## Noise Generator

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. But your mother doesn’t need to know!

| **Input** | **Functionality [OFF / ON]** | **Remark** |
|---|---|---|
| Green Switch | Geiger mode / Noise mode | |

| | | |
|---|---|---|
| Blue Switch | Animated / Counter | |
| Dial | sensitivity | |

### Geiger Counter Mode

| **Input** | **Function** | **Comment** |
|---|---|---|
| Red Button | None | Contact uncle Lode if you have a functionality idea |
| Green Button | None | Contact uncle Lode if you have a functionality idea |
| Blue Button | None | Contact uncle Lode if you have a functionality idea |
| Yellow Button | Temporary sharp increase of tick probability | When pressed, it looks like there is a lot more radiation.<br>**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.

| **Input** | **Functionality** | **Comment** |
|---|---|---|
| Yellow Switch | Prolonged beep | Every beep will sound until another beep takes over if ON. |
| Red Button | Set Lower frequency | Hold while rotating the dial. This is the lowest frequency that will be randomly chosen. |
| Green Button | Upper frequency | Hold while rotating the dial. This is the highest frequency that will be randomly chosen. |
| Blue Button | Tone length | Hold while rotating the dial. Tone length in milliseconds. |
| Yellow Button | Trigger manually | Hold while rotating the dial to trigger manually |

## Whack a mole, reaction game

### Menu

| **Input** | **Functionality [OFF / ON]** | **Remark** |
|---|---|---|

| | | |
|---|---|---|
| Green Switch | Whack a mole / whack a bird | Can only be changed in Menu |
| Blue Switch | Normal mode / Endurance mode | Can only be changed in Menu |
| Yellow Switch | 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 Yellow Switch 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 Yellow Switch 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 Yellow Switch 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 Yellow Switch 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 Says Game

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

### Main menu

| **Button** | **Functionality [OFF / ON]** | |
|---|---|---|
| Green Switch | Simon / without Simon | Normally Simon decides the sequence. In games without Simon, every level, a player adds a step to the sequence. |
| Blue Switch | 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. |
| Yellow Switch | Main menu / Play | When playing, set to OFF to enable menu and reset game |
| Dial | set number of players | |

### Game selection

| **Green Switch** | **Blue Switch** | **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**<br>**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.**<br>**→ 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.<br>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.<br>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]<br>Use momentary buttons to input a sequence.<br>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.<br>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

| **Input** | **Functionality [OFF / ON]** | **Remark** |
|---|---|---|
| Green Switch | Guitar hero / Hex hero | |
| Blue Switch | Normal / Extra mode | Extra mode:<br>Guitar hero: Adds random paused<br>Hex Hero: Input the complement of what you see for all digits with a dot. e.a. 6. → 1001 |
| Yellow Switch | 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 Yellow Switch 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 Yellow Switch 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 Yellow Switch 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.<br>In extra mode (Blue Switch 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 Yellow Switch 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** |
|---|---|---|---|---|---|---|---|
| | | | Red Button | Green Button | Blue Button | Yellow Button | |
| 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 |

# Quizmaster

Once the quizmaster has asked a question. Players can then press a button if they know the answer. It's important to press your button quicker than your opponent. The Luciebox figures out who pressed first.
In independent mode, this app becomes an addictive multiplayer reaction game without a quizmaster asking questions

| **Input** | **Functionality [OFF / ON]** | **Remark** |
|---|---|---|
| Green Switch | Normal / Long maximum wait time | • normal: max 3 seconds<br>• long: max 20 seconds (which is very long!) |
| Blue Switch | 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. |
| Yellow Switch | Question asked | Quizmaster "just asked a question button"<br>In independent mode: ON=play game |
| Dial | | |

## Quizmaster Tutorial

| **Step** | **Comments** |
|---|---|
| 1 | Every player gets a momentary button assigned. |

| | |
|---|---|
| 2 | Quizmaster asks question |
| 3 | Quizmaster sets Yellow Switch 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 Yellow Switch 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 Yellow Switch 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 Yellow Switch |

## Movie Player

Combine animations with songs to create full featured movies.

| **Functionality** | **Switch** | | | | **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 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 some break time.

### Menu

| **Input** | **Functionality [OFF / ON]** | **Remark** |
|---|---|---|
| Green Switch | Enable Break time and auto restart | |
| Blue Switch | 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. |

| | | |
|---|---|---|
| Yellow Switch | Switch ON to start timer | Once started the settings will be saved to the permanent eeprom memory, to be retrieved later. |
| dial | if no Red Button or Green Button is pressed, will set the set POMODORO time | The official Pomodoro time is 25 minutes. |
| Red Button + dial | Hold and turn Dial to set POMODORO break duration. | The official Pomodoro break time is 5 minutes. |
| Green Button + dial | Hold and turn Dial to set random beep probability interval. | Must be bigger than zero to enable.<br>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

| **Input** | **Functionality [OFF / ON]** | **Remark** |
|---|---|---|
| Green Switch | enable/ disable break | |
| Blue Switch | Display Visual timer. | |
| Yellow Switch | 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. |
| Red Button | See number of NO’s | |
| Green Button | See number of Yes’s | |
| Blue Button | add 1 to NO | |
| Yellow Button | add 1 to YES | |

### Performance meter

When enabled, the Pomodoro timer will beep at random intervals during timing. The beeps keep you focussed. The idea is to press the Blue or the Yellow button at each beep. Blue for “I admit, I am procrastinating”, Yellow for “yes I am working”. A tally is kept of the key presses, so you can see your good vs bad ratio with the Red and Green Button. This is the computer equivalent of your mother checking up on you at unexpected moments!

### Manual counter

The buttons can be pressed at any time. So they can also serve 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

# MultiTimer

A chess clock for up to four players.

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Yellow Switch | Menu / Timing | Switching to OFF will always reset the timer |

## Menu

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Green Switch | Fischer Timer | ● Set the time that  will be added to every timer after its turn is finished.<br>● 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.<br>● As copied from boardgamearena.com: When playing a timer's time will never exceed its initial time. |
| Blue Switch | Random | ● A random timer will be chosen at game startup.<br>● When selected, all enabled timer button lights are blinking to indicate a random starter. |
| Yellow Switch | Menu / Start Timing | |
| Dial | Set number of timers,  Set time | ● Normal: Set the number of timers.<br>● 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.<br>● If it is blinking, it is the starting player.<br>● To set the time of the player, press the button while rotating the dial. |

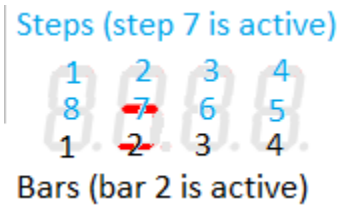
## Timing mode

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Green Switch | None | |
| Blue Switch | Unpause / Pause | |
| Yellow Switch | Menu / Start Timing | |

| | | |
|---|---|---|
| Momentary_buttons | Next timer | • If blinking slow: Timer is selected and ticking. Press the button to select the next timer<br>• If solid: Timer is not selected and not elapsed<br>• If fast blinking in pause mode: Selected but paused timer<br>• If fast blinking not in pause mode: checking remaining time while timer not selected<br>• 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.

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Green Switch | Normal mode / Eeprom Menu | |
| Yellow Switch | Play mode / Program Mode | (Green Switch must be off) |

### Program Mode

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Blue Switch | Single bar / All bars | If All Bars is ON: When assigning a note to a step, it will be assigned to the step of all 4 bars. This is VERY HANDY.<br>Tip: Use it to create a "base" pattern that is the same for all 4 bars before changing steps individually in all bars to make the song interesting. |
| Red Button | Play and show note at active sequencer step | |

| | | |
|---|---|---|
| Green Button | Set and listen to active note without assigning it | Hold button and rotate dial to choose the active note. Keep on holding the button while pressing the red button to assign the selected note to the current active sequencer step. |
| Blue Button | Previous step | |
| Yellow Button | Next step | |
| Dial | Move step | |

### Play Mode

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Red Button | Hold to show assigned notes at active step as they are played | BONUS: hold and rotate dial to transpose the programmed sequence in auto play mode |
| Blue Button | Assign active note to active slot | |
| Yellow Button | Next step | Hold down and rotate dial to quickly go forward |
| Dial | Set step speed | |

### Eeprom menu

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Yellow Switch | LOAD / SAVE | Should a song be saved or loaded? |
| Green Button | Execute | Will save to eeprom or load from eeprom (depending on Yellow Switch) |
| 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.

| **Input** | **Functionality [OFF / ON]** | **Comments** |
|---|---|---|
| Green Switch | Compose mode / Eeprom menu | Compose a song or save or load a song from Eeprom |
| Blue Switch | Note mode / Index mode | Display note value / Display note position in song |

| | | |
|---|---|---|
| Dial | in autoplay: set speed,<br>in manual mode: scroll through all notes. | |

## Compose mode

| **Input** | **Functionality [OFF / ON]** | |
|---|---|---|
| Red Button in Note mode | Play active note | |
| Green Button in Note mode | Select note | Hold down and turn dial to select active note from scale |
| Press Red Button while holding Green Button in Note mode | Program note | It's important to FIRST hold and keep holding Green Button, and then, to program, shortly press Red Button |
| Red Button in Index Mode | Delete current note in song | Delete slot. Move all notes so there is no gap |
| Green Button 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. |
| Blue Button | Previous note in song | |
| Yellow Button | Next note in song | |
| Yellow Switch | Manual / Auto | If ON: the note sequence is played automatically |
| Dial | ● Auto play: set speed<br>● Manual play: scroll through song..<br>● With Green Button in note mode: select note | |

## Eeprom menu

| **Input** | **Functionality [OFF / ON]** | |
|---|---|---|
| Yellow Switch | SAVE / LOAD | Should a song be saved or loaded |
| Red Button | Execute | Save to eeprom or load from eeprom (depending on Yellow Switch) |
| Dial | Choose program number | |

# Uncle Lode’s crazy level

## Hack Time

Observe 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's highscores? If things go wrong, just switch the box on and off again.

Note: The active address is the most left digit on the display

### Modes

| **Input** | **Function [OFF / ON]** | **Remark** |
|---|---|---|
| Green Switch | Address mode / normal mode | |
| Blue Switch | Mute button | Sound OFF / ON.<br>Every byte encodes for a note. On every change, this note is played. Be prepared for post-modernist masterpieces in auto-mode. |
| Yellow Switch | Auto scroll / Manual | |
| Blue Button | HOLD and rotate Dial to scroll through memory in increments of 255 addresses | |
| Yellow Button | HOLD and rotate Dial to scroll through memory in increments of 4 addresses | 4 is chosen, because the display can hold four bytes at its most. |
| Dial | | ● Auto scroll: Change speed<br>● Manual: Change address |

### Address mode

| **Input** | **Functionality** | **Screen** |
|---|---|---|
| Red Button | Set memory type | Cycle through:<br>● FLASH: displays F<br>● RAM: displays R<br>● EEPROM: displays E |

### Normal mode

| **Input** | **Functionality** | **Screen** |
|---|---|---|
| Green | | Cycle through: |

| | | |
|---|---|---|
| Button with dial | | • values displayed as ASCII-like character<br>• values displayed as bytes (7 segments + decimal point = 8 bits represented)<br>• most left digit on display: decimal<br>• most left digit on display: hexadecimal |
| Dial | while Green Button 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

Display the raw byte value. See addendum for byte to display conversion.

### Character representation

Display the byte value as a character. See addendum for byte to character conversion.

## 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<br>● 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!)<br>● byte 0..11: Whack a mole<br>● byte 12..23: Whack a mole endurance mode<br>● byte 24..35: Whack a bird<br>● byte 36..47: Whack a bird endurance mode<br>● byte 48..59: Guitar Hero<br>● byte 60..71: Guitar Hero with stuttering<br>● byte 72..83: Hex Hero<br>● byte 84..95: Hex Hero with stuttering |
| 104..109 | 68..6D | 6 | MultiTimer | ● byte 0: Number of enabled timers<br>● byte 1..4: Init timer per timer [seconds]<br>● byte 5: Fischer time [seconds] |
| 110..112 | 6E..70 | 3 | Pomodoro | ● byte 0: Init time [seconds]<br>● byte 1: Pause time [seconds]<br>● 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 saved Pictures | 4 bytes per picture* 95 pictures |
| 1021..1023 | 3FD..3FF | 3 | Free | Free space. All for you to store secret information, passwords, bitcoin passwords,... |

# Extra features

## Luciebox ON time

This functionality has moved to the second chronometer of the chronometer app.

Nothing can be done in this app. The total time since the Luciebox was switched on is displayed. Switch the box OFF and back ON to reset.

## Inactivity timer

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

## Randomness fun

Computers have a hard time with random value. They produce always the same sequence of values. By “seeding” a random generator, a starting point in that sequence is chosen.

For the Lucie box, the random seed is done with the internal timer at each app initialization. So, at startup, as long as no app has been changed, the same values will show up for generated random events.

In reality: As long as the selector dial was not rotated, the same random sequence will show up when restarting the Luciebox. That means rolling the dice will have result in the same value sequences, the sequencer generator will create the exact same song, guitar hero will show the same blocks,... .

I did this because it’s fun. For better randomness, change apps with the selector dial after switching the Luciebox on.

https://rheingoldheavy.com/better-arduino-random-values/

## Test and Settings

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!

| **Input** | **Function [OFF / ON]** | **Remark** |
|---|---|---|
| Red Switch | Button test | `display: -` |
| Green Switch | Button test | `display:  -` |
| Blue Switch | Button test | `display:   -` |
| Yellow Switch | Button test | `display:    -` |
| Red Button | Button test | `display: 0` |
| Green Button | Button test | `display:  0` |

| | | |
|---|---|---|
| Blue Button | Button test | `display:  0` |
| Yellow Button | Button test | `display:  0` |
| Dial | Settings Menu | Twist back and forth 3 times to activate the settings menu. Once activated, every consecutive twist will activate the next menu item. |

## Settings Menu items

| **Dial twists** | **Menu item** | **Functionality** |
|---|---|---|
| 3 | Enable sound | With Red Button, toggle the sound. This will be stored in the retentive memory. |
| 4 | Battery level | Check the battery or usb voltage. Displayed in millivolts. The battery voltage goes down during the battery life until a level (about 2.7V) until a level where the box cannot operate any longer. Which is called “brown out”.<br><br>Note: The measured voltage can vary from the battery voltage because of an inline Schottky diode (the voltage of a set of new batteries is too high for the microcontroller. The diode lowers the voltage to be always in range for 4 AA batteries. Add about 700mV to the displayed value for battery voltage. |
| 5 | A0 | The value of the selector dial.<br>**The selector dial can be rotated in the A0 menu to check the values without changing apps.**<br>Expected values (estimations): 82/168/253/339/425/510/596/682/767/853/938/1023 (which makes for a difference of 85 per position). (If the selector wiper does not make contact, it shows a low value (~ 50). The analog input is then basically floating.) |
| 6 | A1 | The value is an addition of pressed latching buttons. Red Switch:~64, Green Switch: ~128, Blue Switch: ~256, Yellow Switch: ~512 |
| 7 | A2 | The value is an addition of pressed momentary buttons. Red Button:~64, Green Button: ~128, Blue Button: ~256, Yellow Button: ~512 |
| 8 | A3 | not used for an analog input |
| 9 | A4 | left floating |
| 10 | Firmware version | Have this number ready when contacting the helpdesk. |
| 11 | QTY | Display the amount of times the Luciebox was switched on and at least one momentary button was pressed. |

| | | |
|---|---|---|
| 12 | Reset | Press M0 and then M3 for game high scores and timer settings reset<br>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! |

## 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 |

# Addendums

## Value representation

### 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 |

## Music note values

The length of the rests is

| **Byte value** | ⅛ note | **Byte value** | ¼ note | **Byte value** | ½ note | **Byte value** | Full note | **Byte value** | Rest values (multiples of ⅛) (DISPLAY VALUE) |
|---|---|---|---|---|---|---|---|---|---|
| **0** | A3_8 | **60** | A3_4 | **120** | A3_2 | **180** | A3_1 | **240** | REST_1_8 (-- 1) |
| **1** | As3_8 | **61** | As3_4 | **121** | As3_2 | **181** | As3_1 | **241** | REST_2_8 (-- 2) |
| **2** | B3_8 | **62** | B3_4 | **122** | B3_2 | **182** | B3_1 | **242** | REST_3_8 (-- 3) |
| **3** | C4_8 | **63** | C4_4 | **123** | C4_2 | **183** | C4_1 | **243** | REST_4_8 (-- 4) |
| **4** | Cs4_8 | **64** | Cs4_4 | **124** | Cs4_2 | **184** | Cs4_1 | **244** | REST_5_8 (-- 5) |
| **5** | D4_8 | **65** | D4_4 | **125** | D4_2 | **185** | D4_1 | **245** | REST_6_8 (-- 6) |
| **6** | Ds4_8 | **66** | Ds4_4 | **126** | Ds4_2 | **186** | Ds4_1 | **246** | REST_7_8 (-- 7) |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **7** | E4_8 | **67** | E4_4 | **127** | E4_2 | **187** | E4_1 | **247** | REST_8_8 (-- 8) |
| **8** | F4_8 | **68** | F4_4 | **128** | F4_2 | **188** | F4_1 | **248** | REST_9_8 (-- 9) |
| **9** | Fs4_8 | **69** | Fs4_4 | **129** | Fs4_2 | **189** | Fs4_1 | **249** | REST_10_8 (--A ) |
| **10** | G4_8 | **70** | G4_4 | **130** | G4_2 | **190** | G4_1 | **250** | REST_11_8 (-- B) |
| **11** | Gs4_8 | **71** | Gs4_4 | **131** | Gs4_2 | **191** | Gs4_1 | **251** | REST_12_8 (-- C) |
| **12** | A4_8 | **72** | A4_4 | **132** | A4_2 | **192** | A4_1 | **252** | REST_13_8 (-- D) |
| **13** | As4_8 | **73** | As4_4 | **133** | As4_2 | **193** | As4_1 | **253** | REST_14_8 (-- E) |
| **14** | B4_8 | **74** | B4_4 | **134** | B4_2 | **194** | B4_1 | **254** | REST_15_8 (-- F) |
| **15** | C5_8 | **75** | C5_4 | **135** | C5_2 | **195** | C5_1 | | |
| **16** | Cs5_8 | **76** | Cs5_4 | **136** | Cs5_2 | **196** | Cs5_1 | **255** | RESERVED (eeprom song stop byte) (-- G) |
| **17** | D5_8 | **77** | D5_4 | **137** | D5_2 | **197** | D5_1 | | |
| **18** | Ds5_8 | **78** | Ds5_4 | **138** | Ds5_2 | **198** | Ds5_1 | | |
| **19** | E5_8 | **79** | E5_4 | **139** | E5_2 | **199** | E5_1 | | |
| **20** | F5_8 | **80** | F5_4 | **140** | F5_2 | **200** | F5_1 | | |
| **21** | Fs5_8 | **81** | Fs5_4 | **141** | Fs5_2 | **201** | Fs5_1 | | |
| **22** | G5_8 | **82** | G5_4 | **142** | G5_2 | **202** | G5_1 | | |
| **23** | Gs5_8 | **83** | Gs5_4 | **143** | Gs5_2 | **203** | Gs5_1 | | |
| **24** | A5_8 | **84** | A5_4 | **144** | A5_2 | **204** | A5_1 | | |
| **25** | As5_8 | **85** | As5_4 | **145** | As5_2 | **205** | As5_1 | | |
| **26** | B5_8 | **86** | B5_4 | **146** | B5_2 | **206** | B5_1 | | |
| **27** | C6_8 | **87** | C6_4 | **147** | C6_2 | **207** | C6_1 | | |
| **28** | Cs6_8 | **88** | Cs6_4 | **148** | Cs6_2 | **208** | Cs6_1 | | |
| **29** | D6_8 | **89** | D6_4 | **149** | D6_2 | **209** | D6_1 | | |
| **30** | Ds6_8 | **90** | Ds6_4 | **150** | Ds6_2 | **210** | Ds6_1 | | |
| **31** | E6_8 | **91** | E6_4 | **151** | E6_2 | **211** | E6_1 | | |
| **32** | F6_8 | **92** | F6_4 | **152** | F6_2 | **212** | F6_1 | | |
| **33** | Fs6_8 | **93** | Fs6_4 | **153** | Fs6_2 | **213** | Fs6_1 | | |
| **34** | G6_8 | **94** | G6_4 | **154** | G6_2 | **214** | G6_1 | | |
| **35** | Gs6_8 | **95** | Gs6_4 | **155** | Gs6_2 | **215** | Gs6_1 | | |
| **36** | A6_8 | **96** | A6_4 | **156** | A6_2 | **216** | A6_1 | | |
| **37** | As6_8 | **97** | As6_4 | **157** | As6_2 | **217** | As6_1 | | |
| **38** | B6_8 | **98** | B6_4 | **158** | B6_2 | **218** | B6_1 | | |
| **39** | C7_8 | **99** | C7_4 | **159** | C7_2 | **219** | C7_1 | | |
| **40** | Cs7_8 | **100** | Cs7_4 | **160** | Cs7_2 | **220** | Cs7_1 | | |
| **41** | D7_8 | **101** | D7_4 | **161** | D7_2 | **221** | D7_1 | | |
| **42** | Ds7_8 | **102** | Ds7_4 | **162** | Ds7_2 | **222** | Ds7_1 | | |
| **43** | E7_8 | **103** | E7_4 | **163** | E7_2 | **223** | E7_1 | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **44** | F7_8 | **104** | F7_4 | **164** | F7_2 | **224** | F7_1 | | |
| **45** | Fs7_8 | **105** | Fs7_4 | **165** | Fs7_2 | **225** | Fs7_1 | | |
| **46** | G7_8 | **106** | G7_4 | **166** | G7_2 | **226** | G7_1 | | |
| **47** | Gs7_8 | **107** | Gs7_4 | **167** | Gs7_2 | **227** | Gs7_1 | | |
| **48** | A7_8 | **108** | A7_4 | **168** | A7_2 | **228** | A7_1 | | |
| **49** | As7_8 | **109** | As7_4 | **169** | As7_2 | **229** | As7_1 | | |
| **50** | B7_8 | **110** | B7_4 | **170** | B7_2 | **230** | B7_1 | | |
| **51** | C8_8 | **111** | C8_4 | **171** | C8_2 | **231** | C8_1 | | |
| **52** | Cs8_8 | **112** | Cs8_4 | **172** | Cs8_2 | **232** | Cs8_1 | | |
| **53** | D8_8 | **113** | D8_4 | **173** | D8_2 | **233** | D8_1 | | |
| **54** | Ds8_8 | **114** | Ds8_4 | **174** | Ds8_2 | **234** | Ds8_1 | | |
| **55** | E8_8 | **115** | E8_4 | **175** | E8_2 | **235** | E8_1 | | |
| **56** | F8_8 | **116** | F8_4 | **176** | F8_2 | **236** | F8_1 | | |
| **57** | Fs8_8 | **117** | Fs8_4 | **177** | Fs8_2 | **237** | Fs8_1 | | |
| **58** | G8_8 | **118** | G8_4 | **178** | G8_2 | **238** | G8_1 | | |
| **59** | Gs8_8 | **119** | Gs8_4 | **179** | Gs8_2 | **239** | Gs8_1 | | |

## Display character Values

The letters and digits follow the ASCII standard. The other characters are custom Luciebox characters.
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 is A, right is DP)** | **ASCII value** | **Remark** |
|---|---|---|---|
| ‘ ‘ | | | Space is not implemented, provide SPACE_FAKE_ASCII. All unrecognized values will resolve 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. This is a SPECIAL CASE. It will randomly toggle its segments 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 | |

37

# Todo and issues

## F305

Please write down all idea’s, suggestions, and bugs.

## F304 changes

Update baby mode. Update songs and music notes handling

## F303 Changes

Settings app: ability to debug selector dial at A0

## 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 Blue Switch and Yellow Switch 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.<br>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.<br>**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<br>done |
| song composer: check if dial rotating with If momentary 1 or 2 is pressed, sets note. | reworked done |
| 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. |