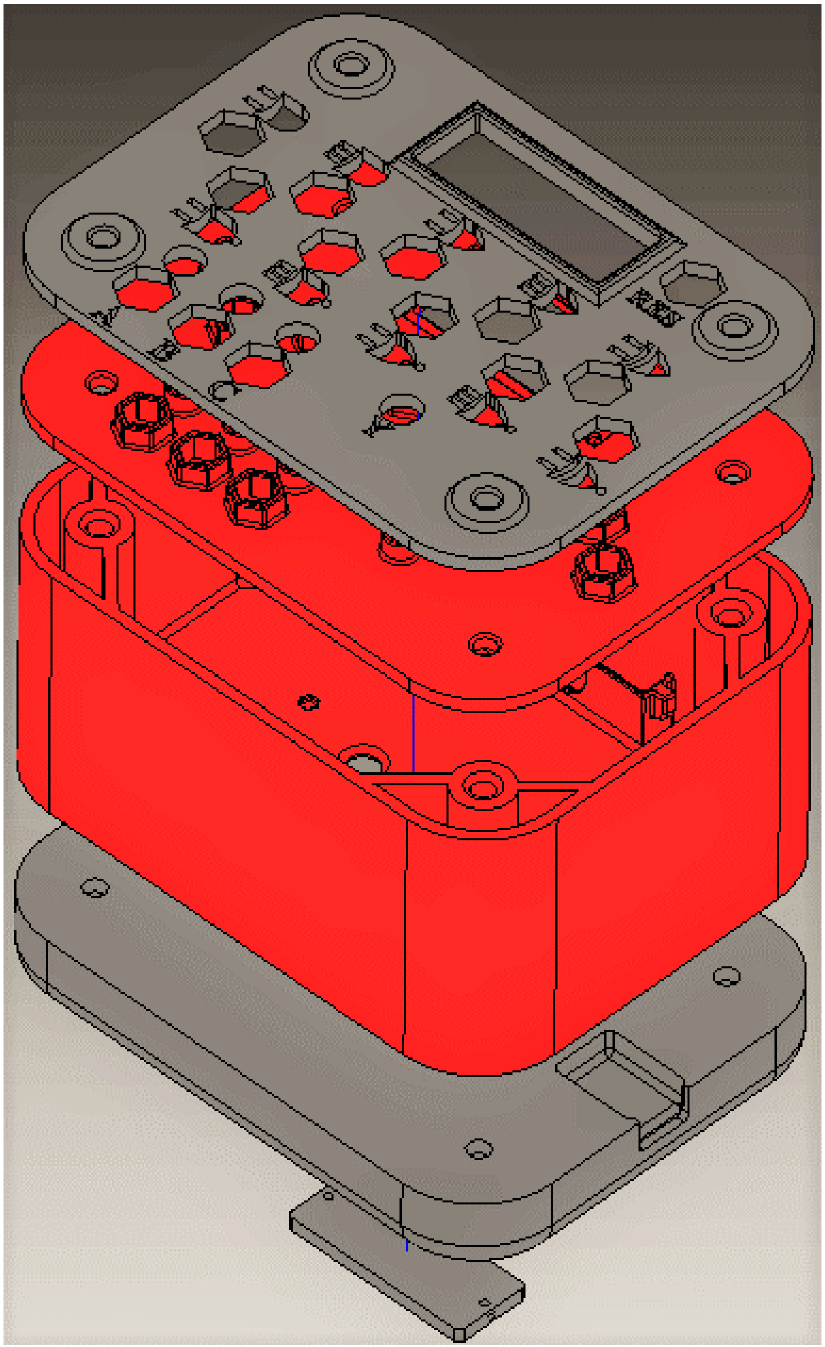
# Name! That! Gate!

*User Manual*



Krietemeyer, B.
2022/05/10

**Contents**:

# Introduction

**Name! That! Gate!** is a logic puzzle & game that tests player knowledge of *Digital Logic Gates*. It's modeled after & is a physical representation of the *ETCAI* (software often used throughout Lake Land College – Mattoon, IL electronic engineering studies) *Digital Challenge* "Name that Gate! Two" game.

The objective is to determine the *Digital Logic Gate* being utilized which creates the specific Output (**Y**) condition (red / green – low / high) given the state of the Inputs (**A**), (**B**), and/or (**C**). There are **140pts possible**, each Stage is worth 14pts, there are x10 Stages, each Input toggled deducts x1pt from Stage points, right *Gate* selection rewards all remaining Stage points, and wrong *Gate* selection deducts all remaining points.

**Gameplay**:

1) From startup, x1 *Digital Logic Gate* (of 10 possible) is randomly loaded
   - *Logic Gates* are: 2-Inputs - AND, NAND, NOR, OR, XOR, XNOR
     3-Inputs - AND, NAND, NOR, OR
2) Inputs always start LOW
   - If only a 2-input *Gate* is loaded then Input (**C**) is deactivated & invalid
3) Initial *Gate* Output (**Y**) reflects what is TRUE for the logic of the loaded *Gate*
4) From this point, the player can either toggle any of the Inputs or select a *Gate*
   - If Input toggled: Even if Input (**C**) invalid, x1 Stage Point deducted if any of those 14pts remain
   - If *Gate* right: All remaining Stage points rewarded and next *Gate* (if any) loaded
   - If *Gate* wrong: All remaining Stage points deducted and next *Gate* (if any) loaded
5) Loaded *Gate* Output (**Y**) will dynamically reflect what is TRUE given the supplied Inputs from (**A**), (**B**), and/or (**C**)
6) Once a *Gate* selection is made the Stage is over & a new *Gate* is randomly loaded
   - No same *Gate* encountered twice during a full (Stage 1 - 10) playthrough
7) After Stage over, Inputs are set LOW again & initial *Gate* Output (**Y**) reflects what is TRUE for that *Gate*.
8) Again, player can either toggle any of the Inputs or select a *Logic Gate*.
9) When no new *Gates* (of 10) remain, the game Ends.

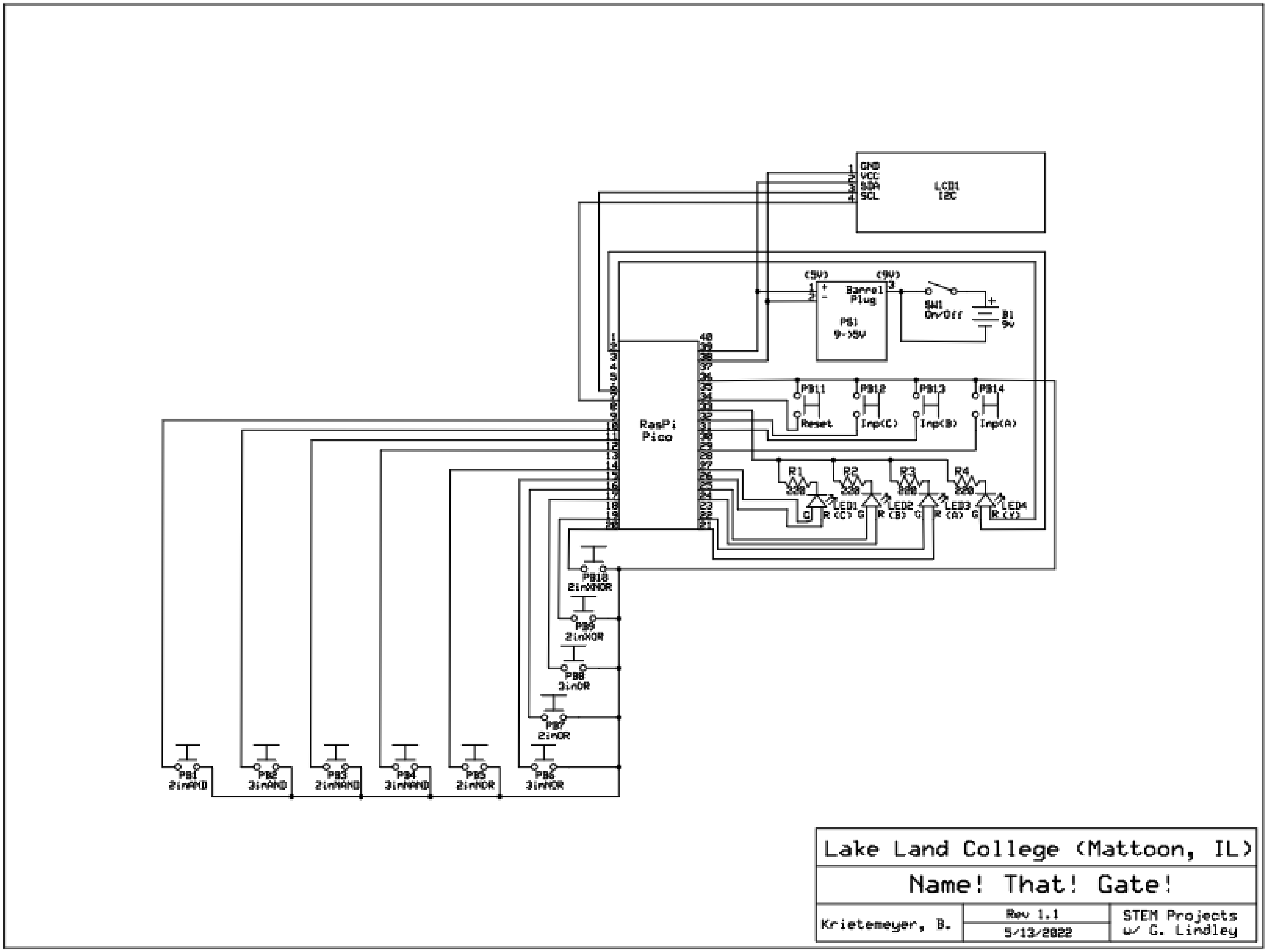
# Special Notes

- When using MicroPython, in order to make code saved to the Raspberry Pi Pico execute upon startup then it must be given the special name: `main.py`
- Protoboard (enclosed) on (4) **Nylon Standoffs** w/ nuts glued into bottom; **fragile**
- DO NOT overtighten any of the (4) **1/4" x 5" Hex Bolts** used to hold enclosure together; just enough to "sandwich" **Power Switch** and make rigid connection w/ all **Bolts** & **Nuts**
- If taking apart remember: **9V Battery** attachment & **Power Switch** wires ran through Base & Middle so DO NOT move too much unless wires disconnected (inside enclosure) to prevent damage
- **LED**s secured by glue (between bottom of LED and recessed hole in Mount)
- **Display** PCB holes were incrementally drilled out in order to fit 3D-printed Mount posts
- **Power Switch** location (3D-print) had to be altered post-print in order to fit hardware
- All enclosed wires connected together by heat shrink & solder combo were applicable
- All connections (male pins to pin headers, female pins to LED posts) reinforced w/ tape
- USB drive w/ all relevant files should be stored inside enclosure
- Thonny IDE used on Windows 10 for programming Raspberry Pi Pico w/ MicroPython
- Helpful websites:
  - https://www.raspberrypi.com/documentation/
  - https://docs.micropython.org/en/latest/library/machine.Pin.html
  - https://simonprickett.dev/buttons-and-leds-with-micropython-for-pi-pico/
  - https://www.tomshardware.com/how-to/lcd-display-raspberry-pi-pico
  - https://forum.micropython.org/viewtopic.php?t=8179

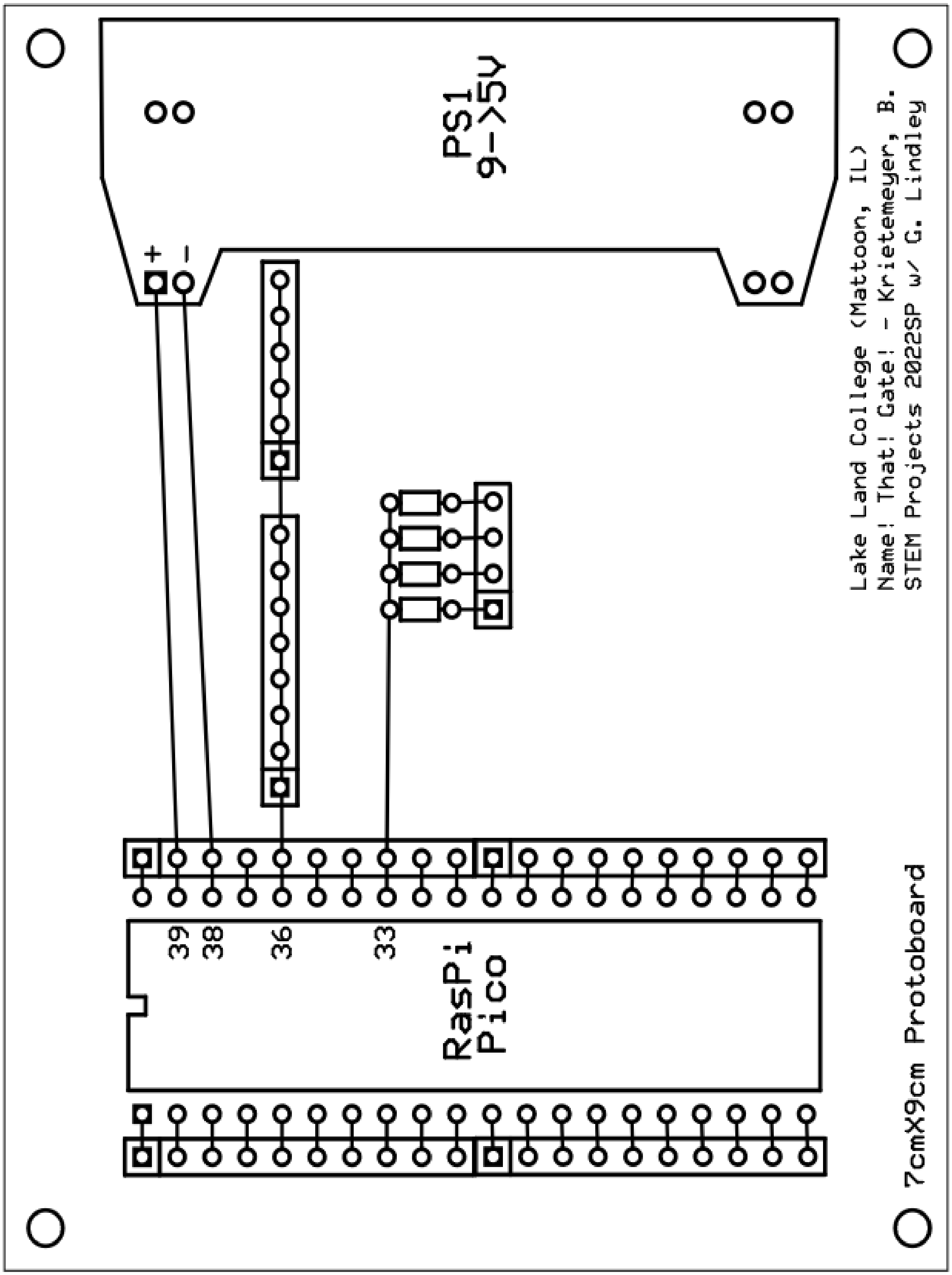
# Parts List

| | |
|---|---|
| 1) Raspberry Pi Pico (2020) | x1 |
| 2) Protoboard (7 x 9cm) | x1 |
| 3) Nylon Standoffs (Protoboard) w/ nuts & screws | x4 |
| 4) Pin Headers (female) | |
| ○ 20-pin | x2 |
| ○ 8-pin | x1 |
| ○ 6-pin | x1 |
| ○ 4-pin | x1 |
| 5) Power Supply (used 9V-to-5V “breadboard” type by “YwRobot”) | x1 |
| 6) 9V Battery | x1 |
| ○ Connector w/ Barrel Plug (Power Supply) | x1 |
| 7) Power Switch | x1 |
| 8) Display (I2C Serial Interface HD44780 LCD) | x1 |
| 9) LEDs (common cathode multicolor RGB) | x4 |
| 10) Resistors (220 Ohm for LEDs) | x4 |
| 11) Pushbuttons (momentary) | |
| ○ White | x1 |
| ○ Yellow | x3 |
| ○ Red | x10 |
| 12) Pin Connectors (male) | |
| ○ Display | x4 |
| ○ LEDs | x12 |
| ○ Pushbuttons | x28 |
| 13) Enclosure (3D-printed) | |
| ○ Base, Middle, Mount, Cover, & Plate | |
| 14) Hex Bolts (1/4” x 5”) w/ Nuts & Washers | x4 |
| 15) Wood Screws (6 x 1/2”) | x2 |
| 16) “Scotch” Bumpers (1/2” ; prevents sliding) | x4 |
| 17) Miscellaneous (Wiring, solder, superglue, etc.) | |

# Schematic


LCD1
I2C
GND
VCC
SDA
SCL
1
2
3
4
(5V)
(9V)
Barrel
Plug
PS1
9->5V
SW1
On/Off
B1
9v
RasPi
Pico
PB11
Reset
PB12
Inp(C)
PB13
Inp(B)
PB14
Inp(A)
R1
220
R2
220
R3
220
R4
220
LED1
R (C)
LED2
R (B)
LED3
R (A)
LED4
R (Y)
G
PB1
2inAND
PB2
3inAND
PB3
2inNAND
PB4
3inNAND
PB5
2inNOR
PB6
3inNOR
PB7
2inOR
PB8
3inOR
PB9
2inXOR
PB10
2inXNOR
Lake Land College (Mattoon, IL)
Name! That! Gate!
Krietemeyer, B.
Rev 1.1
5/13/2022
STEM Projects
w/ G. Lindley

# Board Layout


PS1
9->5V
+
-
39
38
36
33
RasPi
Pico
Lake Land College (Mattoon, IL)
Name! That! Gate! - Krietemeyer, B.
STEM Projects 2022SP w/ G. Lindley
7cmX9cm Protoboard