



Αποτέλεσμα ΙΟ-1

Οδηγός για Minecraft Pi & Physical Computing Blocks

Παραδοτέο: ΙΟ1-Α2: Physical Computing Blocks



STEM4CLIM8

29 Σεπτεμβρίου, 2021

HeartHands Solutions LTD

Authored by: HeartHands Solutions LTD

Project Number: 2020-1-UK01-KA201-079141



Co-funded by the
Erasmus+ Programme
of the European Union

Η υποστήριξη της Ευρωπαϊκής Επιτροπής στην παραγωγή της παρούσας έκδοσης δεν συνιστά αποδοχή του περιεχομένου, το οποίο αντικατοπτρίζει αποκλειστικά τις απόψεις των συντακτών, και η Επιτροπή δεν μπορεί να αναλάβει την ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν.

REVISION HISTORY

Version	Date	Author	Description	Action	Pages
[..]	DD/MM/YYYY	PARTNER ORGANIZATION	[Creation/Insert/ Delete/Update of the document]	[C/I/D/U]	[No. of pages]
1.0	28/06/2021	HESO	Creation of document	C	20
1.1	27/09/2021	HESO	Update of document	U	24

(*) Action: C = Creation, I = Insert, U = Update, R = Replace, D = Delete

REFERENCED DOCUMENTS

ID	Reference	Title
1	2020-1-UK01-KA201-079141	STEM4CLIM8 Proposal
2		

APPLICABLE DOCUMENTS

ID	Reference	Title
1		
2		

ΠΕΡΙΛΗΨΗ ΤΩΝ ΚΥΡΙΟΤΕΡΩΝ ΣΗΜΕΙΩΝ

Σε πρόσφατη έρευνα του ΟΟΣΑ (ΟΟΣΑ- Εκπαίδευση και δεξιότητες σήμερα 2018), που καλύπτει 25 ευρωπαϊκές χώρες, σχεδόν όλες οι χώρες αναφέρουν ελλείψεις δεξιοτήτων που οι εκπαιδευτικοί χρειάζονται για να καλύψουν τις σχολικές ανάγκες, σε συνδυασμό με δυσκολίες στην ενημέρωση των δεξιοτήτων των εκπαιδευτικών. Ωστόσο, η ψηφιοποίηση αναμένεται να αλλάξει ριζικά τον τρόπο που μαθαίνουμε και εργαζόμαστε. Πολλά παιδιά που ξεκινούν το σχολείο σήμερα είναι πιθανό να καταλήξουν να εργάζονται σε θέσεις εργασίας που δεν υφίστανται την παρούσα χρονική στιγμή. Η προετοιμασία των μαθητών για αυτές τις νέες προκλήσεις σημαίνει ότι όχι μόνο πρέπει να διασφαλίσουμε ότι έχουν τις σωστές τεχνικές δεξιότητες, αλλά και να ενισχύσουμε τις συναισθηματικές και κοινωνικές τους δεξιότητες. Η ανθεκτικότητα, η ατομική ικανότητα να ξεπεραστούν οι δυσμενείς περιστάσεις και να χρησιμοποιηθούν ως πηγές για προσωπική ανάπτυξη, βρίσκεται στον πυρήνα της επιτυχούς προσαρμογής στην αλλαγή και, συνεπώς, στην ενεργή συμμετοχή στον ψηφιακό κόσμο. Ταυτόχρονα, πρέπει να αναγνωρίσουμε τον εθισμό στο Διαδίκτυο και τις συμπεριφορές που οδηγούν στον εθισμό ως αναδυόμενα προβλήματα για τη νεολαία μας. Μια προσέγγιση STEM που γεφυρώνει τον φυσικό υπολογισμό με περιβαλλοντική συνείδηση ενώ εστιάζει σε συνεργατικές δραστηριότητες εκτός οθόνης είναι ένας εξαιρετικός τρόπος βελτίωσης των τεχνικών ικανοτήτων, ενώ ενισχύει τις συναισθηματικές και κοινωνικές δεξιότητες.

Το STEM4CLIM8 έχει ως πρωταρχικό στόχο να παράγει προσεγγίσεις και εργαλεία για να βοηθήσει εκείνους που εργάζονται με παιδιά να επικοινωνήσουν μαζί τους για να τους βοηθήσουν να ασχοληθούν με τον προγραμματισμό και να αναπτύξουν δεξιότητες που σχετίζονται με το STEM. Στόχος του προγράμματος είναι να το επιτύχει όχι αυξάνοντας τον χρόνο παραμονής στην οθόνη, αλλά ενθαρρύνοντας το παιχνίδι μέσω της δημιουργίας ενός προσαρμοσμένου εικονικού κόσμου χρησιμοποιώντας το Minecraft για την εκτέλεση αποστολών που ασχολούνται με φυσικές καταστροφές και τη χρήση διαδραστικών αντικειμένων που θα προγραμματιστούν για να αλληλοεπιδράσουν με το κόσμο του Minecraft μέσω του Raspberry GPIO. Οι αποστολές θα αποκαλύψουν την επιστήμη πίσω από τα φυσικά φαινόμενα που συνδέονται συχνά με την κλιματική αλλαγή και θα εμπνεύσουν την περιβαλλοντική συνείδηση ενώ ταυτόχρονα θα ενισχύσουν τις δεξιότητες STEM.

Αναφορά:

*OECD- Education and Skills Today, Succeeding with resilience-Lessons for schools, January 29, 2018, Retrieved February 18, 2021 from: <https://oecdutoday.com/succeeding-with-resilience-lessons-for-schools/>

Πίνακας περιεχομένων

ΠΕΡΙΛΗΨΗ ΤΩΝ ΚΥΡΙΟΤΕΡΩΝ ΣΗΜΕΙΩΝ	3
1. PHYSICAL COMPUTING BLOCKS ΣΤΟ MINECRAFT PI.....	5
1.1 ΕΓΚΑΤΑΣΤΗΣΤΕ ΤΗΝ “FISHING TOWN” ΣΤΟΝ ΚΟΣΜΟ ΤΟΥ MINECRAFT PI.....	5
1.2 ΣΕΝΑΡΙΑ ΚΑΙ ΑΛΛΗΛΕΠΙΔΡΑΣΕΙΣ.....	6
1.3 ΣΕΙΣΜΟΣ	6
1.3.1 Εξοπλισμός και σενάριο	6
1.3.2 Ηλεκτρονικό κύκλωμα and physical block	7
1.3.3 Προγραμματισμός με Python.....	9
1.4 ΥΠΟΧΩΡΗΣΗ ΕΔΑΦΟΥΣ	10
1.4.1 Εξοπλισμός και σενάριο	10
1.4.2 Ηλεκτρονικό κύκλωμα και physical block	10
1.4.3 Προγραμματισμός με Python.....	12
1.5 ΕΚΡΗΣΗ ΗΦΑΙΣΤΕΙΟΥ	13
1.5.1 Εξοπλισμός και σενάριο	13
1.5.2 Ηλεκτρονικό κύκλωμα και physical block	14
1.5.3 Προγραμματισμός με Python.....	16
1.6 HEATWAVE.....	17
1.6.1 Equipment and scenario.....	17
1.6.2 Electronic circuit and execution of activity.....	17
1.6.3 Python programming	20
1.7 FLOOD.....	21
1.7.1 Equipment and scenario.....	21
1.7.2 Electronic circuit and execution of activity.....	22
1.7.3 Python programming	25
1.7 CONCLUSION	25
REFERENCES	25

1. Physical Computing Blocks στο Minecraft Pi

Ένα σύνολο Physical Computing Blocks για να τα συναρμολογήσετε μόνοι σας έχει σχεδιαστεί για να αλληλοεπιδρά με ένα κόσμο Minecraft στην έκδοση του Raspberry Pi όπου παρουσιάζει μια σειρά από φυσικές καταστροφές. Αυτά τα μπλοκ θα συνδεθούν μέσω του Raspberry Pi GPIO και θα προγραμματιστούν μέσω της γλώσσας προγραμματισμού Python προκειμένου να αλληλοεπιδρούν με τον κόσμο του Minecraft Pi, προσομοιώνοντας τις φυσικές καταστροφές και τα καταστροφικά αποτελέσματά τους σε μια μικρή πόλη που ονομάζεται «Fishing Town (Ψαρό πόλη)».

Αυτός ο οδηγός, που απευθύνεται σε εκπαιδευτικούς, προσφέρει οδηγίες για τη συναρμολόγηση των Physical Computing Blocks, για τη σύνδεσή τους στο GPIO και για τον προγραμματισμό τους χρησιμοποιώντας τη γλώσσα προγραμματισμού Python και το παιχνίδι Minecraft στο Raspberry Pi.

Οι δάσκαλοι πρέπει να είναι εξοπλισμένοι με τα ακόλουθα:

- Την STEM4CLIM8 κονσόλα
- Τα Physical Computing Blocks
- Το εκπαιδευτικό υλικό για τη διδασκαλία αυτών των φαινομένων.

1.1 Εγκαταστήστε την "Fishing Town" στον κόσμο του Minecraft Pi

Για να εκτελέσετε οποιοδήποτε από τα σενάρια φυσικών καταστροφών, πρέπει πρώτα να εγκαταστήσετε έναν χάρτη μιας ψαροπόλεως που θα χρησιμοποιηθεί ως το πεδίο δοκιμών των διαφόρων καταστροφών. Για να το κάνετε αυτό, πρέπει πρώτα να ενεργοποιήσετε το Raspberry Pi στην STEM4CLIM8 κονσόλα.

Στην επιφάνεια εργασίας, θα δείτε έναν φάκελο με το όνομα "Fishing Town Map". Κάντε διπλό κλικ πάνω του για να δείτε τα περιεχόμενα του φακέλου. Στη συνέχεια, ακολουθήστε αυτά τα απλά βήματα:

1. Ανοίχτε το Minecraft Pi και δημιουργήστε έναν νέο κόσμο, μετά κλείστε το Minecraft Pi
2. Ανοίχτε το "File Manager".
3. Πατήστε στο "View" και επιλέξτε "Show Hidden".
4. Κάντε διπλό κλικ στον φάκελο με το όνομα ".minecraft".
5. Κάντε διπλό κλικ στον φάκελο με το όνομα "games".
6. Κάντε διπλό κλικ στον φάκελο με το όνομα "com.mojang".
7. Κάντε διπλό κλικ στον φάκελο με το όνομα "minecraftWorlds".
8. Αντιγράψτε τα περιεχόμενα του φακέλου "Fishing Town Map" σε έναν υπάρχοντα φάκελο "world".
9. Μετονομάστε αυτόν τον φάκελο σε "fishingtown".
10. Ανοίξτε το Minecraft Pi και φορτώστε τον κόσμο "fishingtown".

Ένας κόσμος Minecraft Pi "Fishing Town" θα φορτωθεί. Λάβετε υπόψη ότι πρέπει να επαναλαμβάνετε αυτή τη διαδικασία κάθε φορά που χρειάζεστε έναν νέο και χωρίς καταστροφές χάρτη.

Αυτός ο κόσμος του Minecraft Pi θα χρησιμοποιηθεί για την προσομοίωση των παρακάτω φυσικών καταστροφών:

- Σεισμός
- Έκρηξη ηφαιστείου
- Υποχώρηση Εδάφους
- Καύσωνας
- Πλημμύρα

1.2 Σενάρια και αλληλεπιδράσεις

Με τη χρήση του Raspberry Pi GPIO, των αισθητήρων, των ηλεκτρονικών, των περιφερειακών και των χαρτοκιβωτίων που θα συνθέσετε μόνοι σας, οι μαθητές θα μάθουν για τις φυσικές καταστροφές, θα δοκιμάσουν τα αποτελέσματά τους και θα συζητήσουν τις συνέπειές τους.

Αρχικά, οι δάσκαλοι θα καθοδηγήσουν τους μαθητές τους για το πώς να συνδέσουν τους αισθητήρες και άλλα ηλεκτρονικά στο Raspberry Pi, πώς να συνθέσουν τα χαρτοκιβώτια και πώς να εκτελέσουν τις δραστηριότητες στο Minecraft Pi μέσω της γλώσσας προγραμματισμού Python.

Ο κόσμος του Minecraft Pi έχει διαμορφωθεί ως μια μικρή ψαράδικη πόλη στην οποία θα συμβούν οι φυσικές καταστροφές. Οι δραστηριότητες κατηγοριοποιούνται σε πέντε θεματικές ενότητες:

1. Σεισμός
2. Έκρηξη ηφαιστείου
3. Υποχώρηση Εδάφους
4. Καύσωνας
5. Πλημμύρα

Για κάθε δραστηριότητα έχουν αναπτυχθεί συγκεκριμένες οδηγίες για τον τρόπο εκτέλεσης. Συνιστάται να χωρίσετε τους μαθητές σας σε πολλές ομάδες και κάθε ομάδα θα πρέπει να έχει πρόσβαση σε τουλάχιστον μία κονσόλα STEM4CLIM8, ένα συνδεδεμένο πληκτρολόγιο και ποντίκι καθώς και ένα ηχείο.

1.3 Σεισμός

1.3.1 Εξοπλισμός και σενάριο

Αυτή η δραστηριότητα θα παρουσιάσει έναν σεισμό στον κόσμο του Minecraft Pi. Οι μαθητές θα συνδέσουν ένα κουμπί στο Raspberry Pi GPIO και θα το προγραμματίσουν για να δημιουργήσει σεισμό μετά από ένα πάτημα.

Η ιδέα είναι να δημιουργηθεί ένα physical computing block χρησιμοποιώντας τα ακόλουθα ηλεκτρονικά και περιφερειακά:

1. 1 x Breadboard
2. 1 x Κουμπί πιεστικό με καπάκι κουμπιού
3. 1 x 220 Ohm Αντίσταση
4. 2 x Καλώδια γεφύρωσης (Jumper) από αρσενικό σε θηλυκό
5. 1 x Κατασκευή χαρτοκιβωτίου

Αφού συνδεθούν όλα, οι μαθητές θα πρέπει να φορτώσουν τον χάρτη "Fishing Town" και το σενάριο Python με το όνομα "earthquake.py" που μπορεί να βρεθεί στην επιφάνεια εργασίας κάθε κονσόλας STEM4CLIM8.

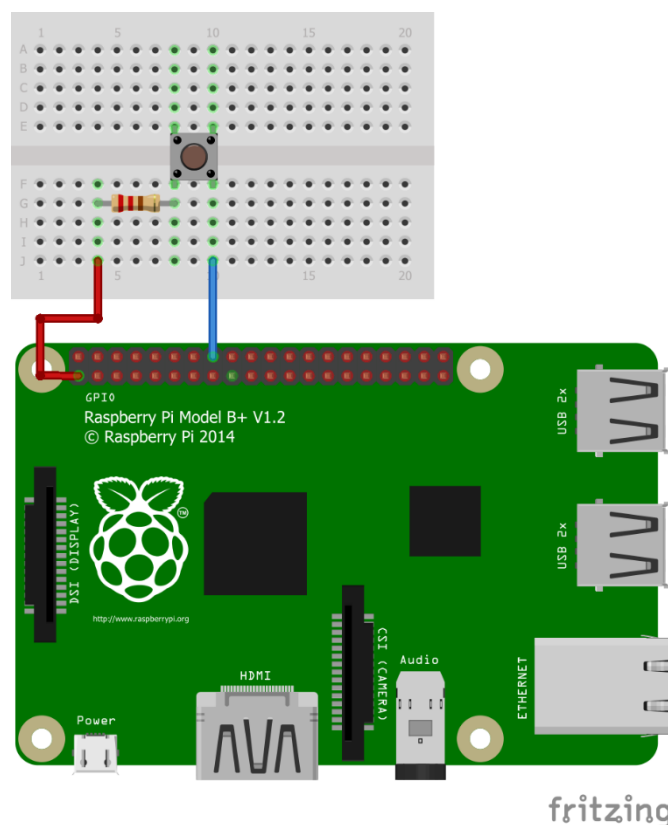
Στη συνέχεια, κάθε φορά που πατάτε το κουμπί, θα συμβεί ένας σεισμός στον κόσμο του Minecraft Pi που θα καταστρέψει κτίρια και άλλα τετράγωνα της "Fishing Town", παρόμοιο με έναν πραγματικό σεισμό υψηλού μεγέθους.

Μετά τη δραστηριότητα θα πρέπει να ακολουθήσει μια συζήτηση με τους μαθητές σας σχετικά με τις επιπτώσεις του σεισμού και την έννοια των σωστά κατασκευασμένων κτιρίων.

1.3.2 Ηλεκτρονικό κύκλωμα and physical block

Η σύνδεση ενός κουμπιού στο GPIO του Raspberry Pi είναι μια πολύ απλή διαδικασία όπως απεικονίζεται στην Εικόνα 1 παρακάτω:

1. Συνδέστε τη δεξιά πλευρά του κουμπιού στην ακίδα εισόδου 16 (GPIO23) χρησιμοποιώντας ένα καλώδιο γεφύρωσης Αρσενικό σε Θηλυκό.
2. Συνδέστε την αριστερή πλευρά του κουμπιού στην αντίσταση 220 Ohm.
3. Συνδέστε την άλλη πλευρά της αντίστασης στην ακίδα 1 3.3V.
4. Το κύκλωμα σας είναι έτοιμο.



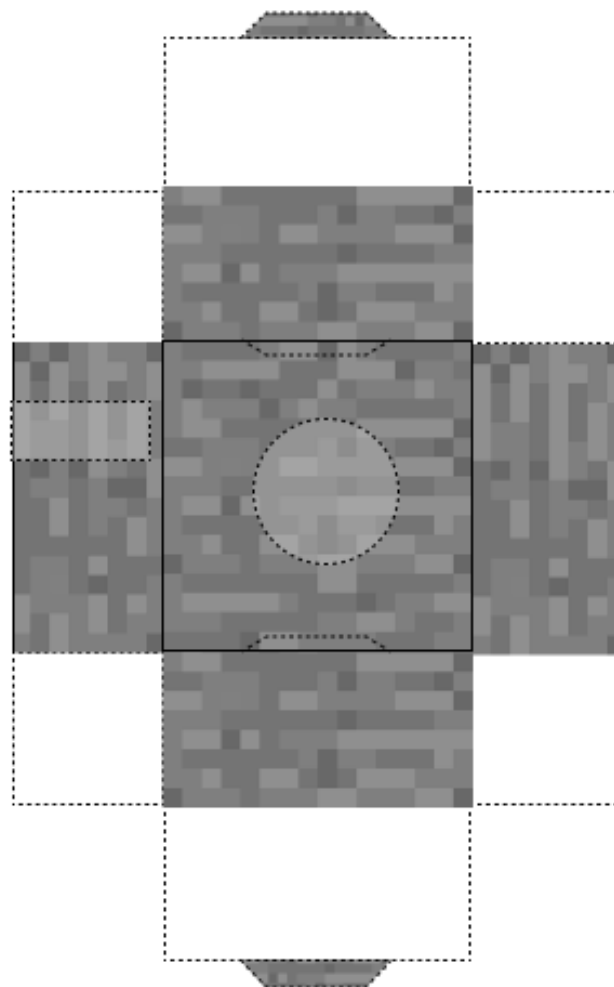
Εικόνα 1 Σύνδεση ενός κουμπιού στο GPIO χρησιμοποιώντας ένα breadboard και δύο καλώδια γεφύρωσης (Jumper).

Κατανοώντας τις διαφορετικές ακίδες GPIO, θα πρέπει να ρίξετε μια ματιά στην Εικόνα 2 που περιγράφει τις ιδιότητες κάθε ακίδας.



Εικόνα 2 Raspberry Pi GPIO Ακίδες

Αφού δημιουργήσετε το κύκλωμα, πρέπει να συναρμολογήσετε το χάρτινο μπλοκ και να τοποθετήσετε το κύκλωμα σε αυτό. Για το σκοπό αυτό, πρέπει να χρησιμοποιήσετε την παρακάτω εκτυπώσιμη εικόνα. Για την καλύτερη χρήση, φροντίστε να εκτυπώσετε την κατασκευή σε λευκό χαρτόνι A4 και, στη συνέχεια, κόψτε τις καθορισμένες άκρες.



Εικόνα 3 Εκτυπώσιμη κατασκευή χαρτοκιβωτίου για physical computing block με κουμπί.

1.3.3 Προγραμματισμός με Python

Απλώς ανοίξτε την Thonny Python στην επιφάνεια εργασίας του Raspberry Pi και, στη συνέχεια, φορτώστε το σενάριο που ονομάζεται "earthquake.py" το οποίο μπορείτε να βρείτε στην επιφάνεια εργασίας ή αντιγράψτε και επικολλήστε το σενάριο απευθείας στον συντάκτη Python της επιλογής σας. Εάν αποφασίσετε να συνδέσετε το κουμπί σε διαφορετική ακίδα GPIO, βεβαιωθείτε ότι έχετε αλλάξει τον αριθμό GPIO στο παρακάτω σενάριο (γραμμές 10 και 42).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minecraft
import mcpi.block as block
mc = minecraft.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

def earthquake(x, z):
    mc.postToChat('Earthquake!')
    y = mc.getHeight(x, z)
    endtime = time.time() + 60
    nearthtime = time.time()
    while endtime > time.time():
        if time.time() > nearthtime:
            nearthtime = time.time() + 5
        ppos = mc.player.getPos()
        if ppos.x < x+100 and ppos.x > x-100:
            if ppos.y < y+100 and ppos.y > -60:
                if ppos.z < z+100 and ppos.z > z-100:
                    mc.player.setPos(ppos.x, ppos.y, ppos.z)
        bx = random.randint(x-100, x+100)
        by = y
        bz = random.randint(z-100, z+100)
        if mc.getHeight(bx, bz) > -50:
            by = mc.getHeight(bx, bz)
        if mc.getBlock(bx, by, bz) in [block.GLASS.id, block.GLASS_PANE.id]:
            mc.setBlock(bx, by, bz, block.AIR.id)
            continue
        mc.setBlock(bx, by, bz, block.GRAVEL.id)
        mc.setBlocks(bx, by-1, bz, bx, -60, bz, block.AIR.id)

disasters = [earthquake]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Press button for earthquake!')
    while True:
        if GPIO.input(23) == GPIO.HIGH: #Look for button press
            t = random.randint(5, 60)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

1.4 Υποχώρηση Εδάφους

1.4.1 Εξοπλισμός και σενάριο

Αυτή η δραστηριότητα θα παρουσιάσει μια υποχώρηση στο έδαφος στον κόσμο του Minecraft Pi που θα συμβεί μετά από έναν σεισμό. Παρόμοια με την πρώτη δραστηριότητα, οι μαθητές θα συνδέσουν ένα κουμπί στο Raspberry Pi GPIO και θα το προγραμματίσουν ώστε να δημιουργήσει έναν σεισμό και την υποχώρηση του εδάφους μετά από ένα πάτημα.

Η δραστηριότητα χρησιμοποιεί τα ακόλουθα ηλεκτρονικά και περιφερειακά:

1. 1 x Breadboard
2. 1 x Κουμπί πιεστικό με καπάκι κουμπιού
3. 1 x 220 Ohm Αντίσταση
4. 2 x Καλώδια γεφύρωσης (Jumper) από αρσενικό σε θηλυκό
5. 1 x Κατασκευή χαρτοκιβωτίου

Αφού συνδεθούν όλα, οι μαθητές θα πρέπει να φορτώσουν τον χάρτη "Fishing Town" και το σενάριο Python με το όνομα "sinkhole.py" που μπορεί να βρεθεί στην επιφάνεια εργασίας κάθε κονσόλας STEM4CLIM8.

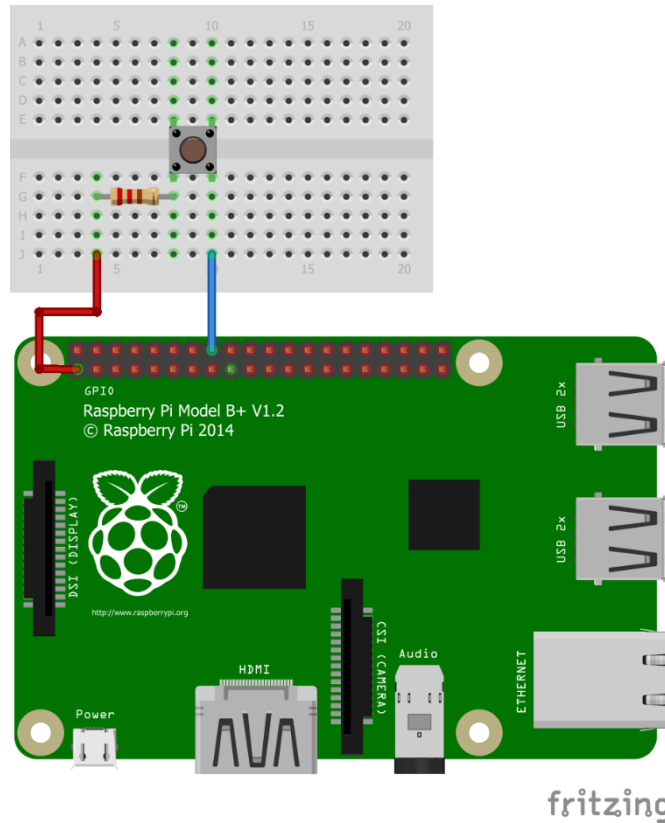
Στη συνέχεια, κάθε φορά που πατάτε το κουμπί, θα συμβεί ένας σεισμός στον κόσμο του Minecraft Pi, ακολουθούμενος από μια υποχώρηση του εδάφους που θα καταστρέψει κτίρια και θα βυθίσει το έδαφος σαν μια πραγματική καταστροφή.

Μετά τη δραστηριότητα θα πρέπει να ακολουθήσει μια συζήτηση με τους μαθητές σας σχετικά με τις επιπτώσεις των σεισμών και τις υποχωρήσεις του εδάφους.

1.4.2 Ηλεκτρονικό κύκλωμα και *physical block*

Η σύνδεση ενός κουμπιού στο GPIO του Raspberry Pi είναι μια πολύ απλή διαδικασία όπως φαίνεται στην Εικόνα 4 παρακάτω:

1. Συνδέστε τη δεξιά πλευρά του κουμπιού στην ακίδα εισόδου 16 (GPIO23) χρησιμοποιώντας ένα καλώδιο γεφύρωσης Αρσενικό σε Θηλυκό.
2. Συνδέστε την αριστερή πλευρά του κουμπιού στην αντίσταση 220 Ohm.
3. Συνδέστε την άλλη πλευρά της αντίστασης στην ακίδα 1 3.3V.
4. Το κύκλωμα σας είναι έτοιμο.



fritzing

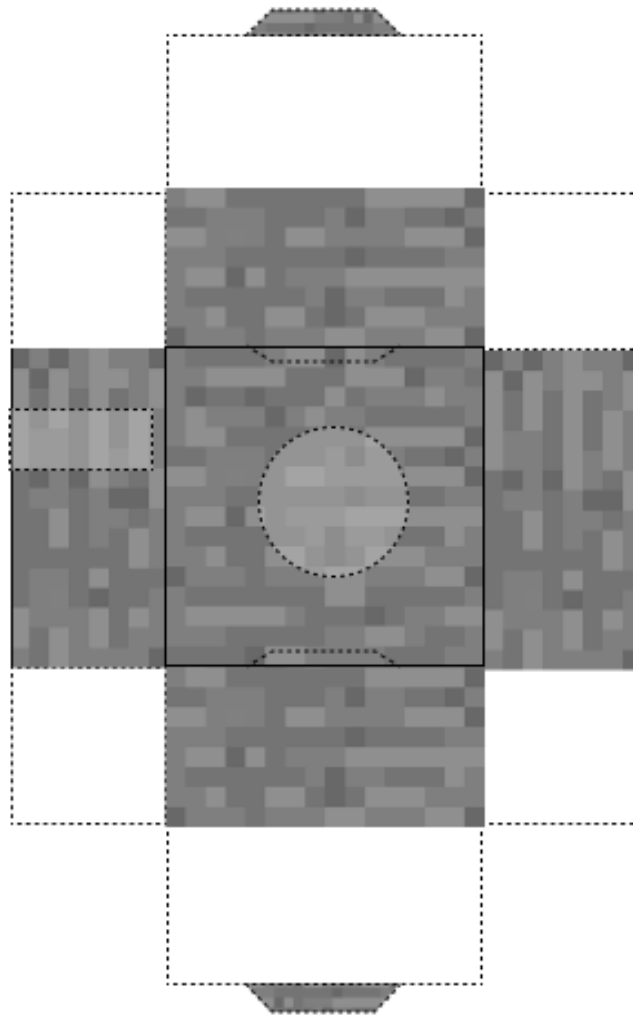
Εικόνα 4 Σύνδεση ενός κουμπιού στο GPIO χρησιμοποιώντας ένα breadboard και δύο καλώδια γεφύρωσης (Jumper).

Κατανοώντας τις διαφορετικές ακίδες GPIO, θα πρέπει να ρίξετε μια ματιά στην Εικόνα 5 που περιγράφει τις ιδιότητες κάθε ακίδας.



Εικόνα 5 Raspberry Pi GPIO Ακίδες

Αφού δημιουργήσετε το κύκλωμα, πρέπει να συναρμολογήσετε το χάρτινο μπλοκ και να τοποθετήσετε το κύκλωμα σε αυτό. Για το σκοπό αυτό, πρέπει να χρησιμοποιήσετε την παρακάτω εκτυπώσιμη εικόνα. Για την καλύτερη χρήση, φροντίστε να εκτυπώσετε την κατασκευή σε λευκό χαρτόνι A4 και, στη συνέχεια, κόψτε τις καθορισμένες άκρες.



Εικόνα 6 Εκτυπώσιμη κατασκευή χαρτοκιβωτίου για physical computing block με κουμπί.

1.4.3 Προγραμματισμός με Python

Απλώς ανοίξετε την Thonny Python στην επιφάνεια εργασίας του Raspberry Pi και, στη συνέχεια, φορτώστε το σενάριο που ονομάζεται "sinkhole.py" το οποίο μπορείτε να βρείτε στην επιφάνεια εργασίας ή αντιγράψτε και επικολλήστε το σενάριο απευθείας στον συντάκτη Python της επιλογής σας. Εάν αποφασίσετε να συνδέσετε το κουμπί σε διαφορετική ακίδα GPIO, βεβαιωθείτε ότι έχετε αλλάξει τον αριθμό GPIO στο παρακάτω σενάριο (γραμμές 10 και 57).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

def sinkhole(x, z):
    mc.postToChat('Earthquake!')
    y = mc.getHeight(x, z)
    endtime = time.time() + 60
    nearthtime = time.time()
```

```

while endtime > time.time():
    if time.time() > nearthtime:
        nearthtime = time.time() + 5
    ppos = mc.player.getPos()
    if ppos.x < x+50 and ppos.x > x-50:
        if ppos.y < y+50 and ppos.y > -60:
            if ppos.z < z+50 and ppos.z > z-50:
                mc.player.setPos(ppos.x, ppos.y, ppos.z)
    bx = random.randint(x-50, x+50)
    by = y
    bz = random.randint(z-50, z+50)
    if mc.getHeight(bx, bz) > -50:
        by = mc.getHeight(bx, bz)
    if mc.getBlock(bx, by, bz) in [block.GLASS.id, block.GLASS_PANE.id]:
        mc.setBlock(bx, by, bz, block.AIR.id)
        continue
    mc.setBlock(bx, by, bz, block.GRAVEL.id)
    mc.setBlocks(bx, by-1, bz, bx, -60, bz, block.AIR.id)

mc.postToChat('Sinkhole!')
blks = []
y = mc.getHeight(x, z)
xdist = random.randint(1, 20)
for bx in range(-xdist, xdist+1):
    zdist = random.randint(1, 20)
    for bz in range(-zdist, zdist+1):
        blks.append([x+bx, z+bz])
for blk in blks:
    mc.setBlocks(blk[0], mc.getHeight(blk[0], blk[1]), blk[1], blk[0], -60, blk[1],
block.AIR.id)
    mc.setBlocks(blk[0], -55, blk[1], blk[0], -60, blk[1], block.LAVA.id)
for blk in blks:
    mc.setBlock(blk[0], y, blk[1], block.GRAVEL.id)

disasters = [sinkhole]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Press button for disaster!')
    while True:
        if GPIO.input(23) == GPIO.HIGH: #Look for button press
            t = random.randint(5, 60)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))

```

1.5 Έκρηξη ηφαιστείου

1.5.1 Εξοπλισμός και σενάριο

Αυτή η δραστηριότητα θα παρουσιάσει μια έκρηξη ηφαιστείου στον κόσμο του Minecraft Pi. Για άλλη μια φορά, οι μαθητές θα συνδέσουν ένα κουμπί στο Raspberry Pi GPIO και θα το προγραμματίσουν για να δημιουργήσει μια ηφαιστειακή έκρηξη μετά από ένα πάτημα.

Για αυτή τη δραστηριότητα θα πρέπει να χρησιμοποιήσετε τα παρακάτω ηλεκτρονικά και περιφερειακά:

1. 1 x Breadboard
2. 1 x Κουμπί πιεστικό με καπάκι κουμπιού
3. 1 x 220 Ohm Αντίσταση
4. 2 x Καλώδια γεφύρωσης (Jumper) από αρσενικό σε θηλυκό
5. 1 x Κατασκευή χαρτοκιβωτίου

Αφού συνδεθούν όλα, οι μαθητές θα πρέπει να φορτώσουν τον χάρτη «Fishing Town» και το σενάριο Python με το όνομα «volcano.py» που μπορεί να βρεθεί στην επιφάνεια εργασίας κάθε κονσόλας STEM4CLIM8.

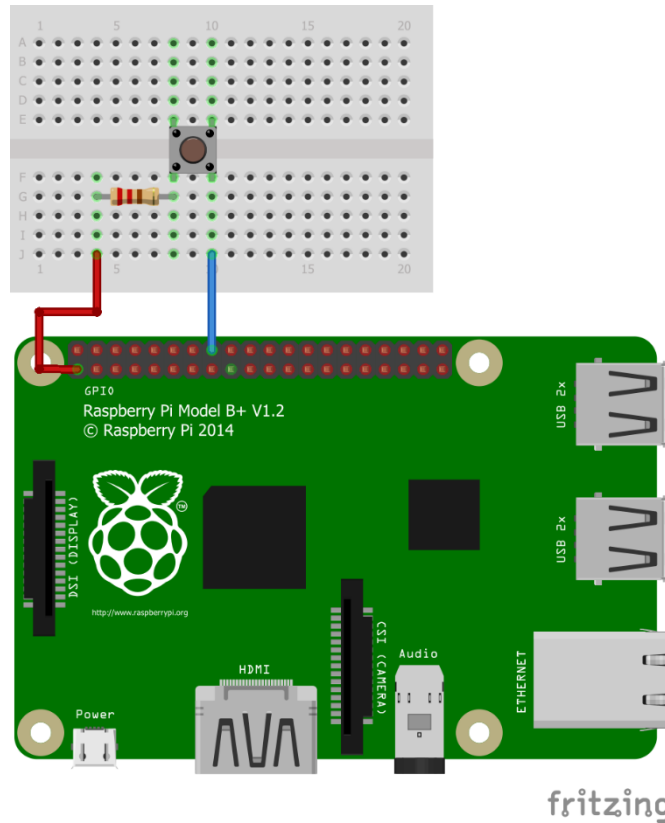
Στη συνέχεια, κάθε φορά που πατάτε το κουμπί, ένα ηφαιστειο θα εμφανίζεται στον κόσμο του Minecraft Pi που θα εκραγεί και η λάβα θα αρχίσει να ρέει προς την «Fishing Town». Μετά από λίγο, η λάβα θα γίνει πέτρα, παρόμοια με μια πραγματική ηφαιστειακή έκρηξη. Κτίρια και άλλα τετράγωνα θα εξαφανιστούν και τα δέντρα και το γρασίδι θα πάρουν φωτιά.

Μετά τη δραστηριότητα θα πρέπει να ακολουθήσει μια συζήτηση με τους μαθητές σας σχετικά με τον αντίκτυπο των κτιρίων και τον υπερ-πληθυσμό των πόλεων, καθώς οι φυσικές καταστροφές δεν έχουν όρια και μπορεί να απειλήσουν τη ζωή και τις συνθήκες διαβίωσής μας.

1.5.2 Ηλεκτρονικό κύκλωμα και physical block

Η σύνδεση ενός κουμπιού στο GPIO του Raspberry Pi είναι μια πολύ απλή διαδικασία όπως φαίνεται στην Εικόνα 7 παρακάτω:

1. Συνδέστε τη δεξιά πλευρά του κουμπιού στην ακίδα εισόδου 16 (GPIO23) χρησιμοποιώντας ένα καλώδιο γεφύρωσης Αρσενικό σε Θηλυκό.
2. Συνδέστε την αριστερή πλευρά του κουμπιού στην αντίσταση 220 Ohm.
3. Συνδέστε την άλλη πλευρά της αντίστασης στην ακίδα 1 3.3V.
4. Το κύκλωμα σας είναι έτοιμο.

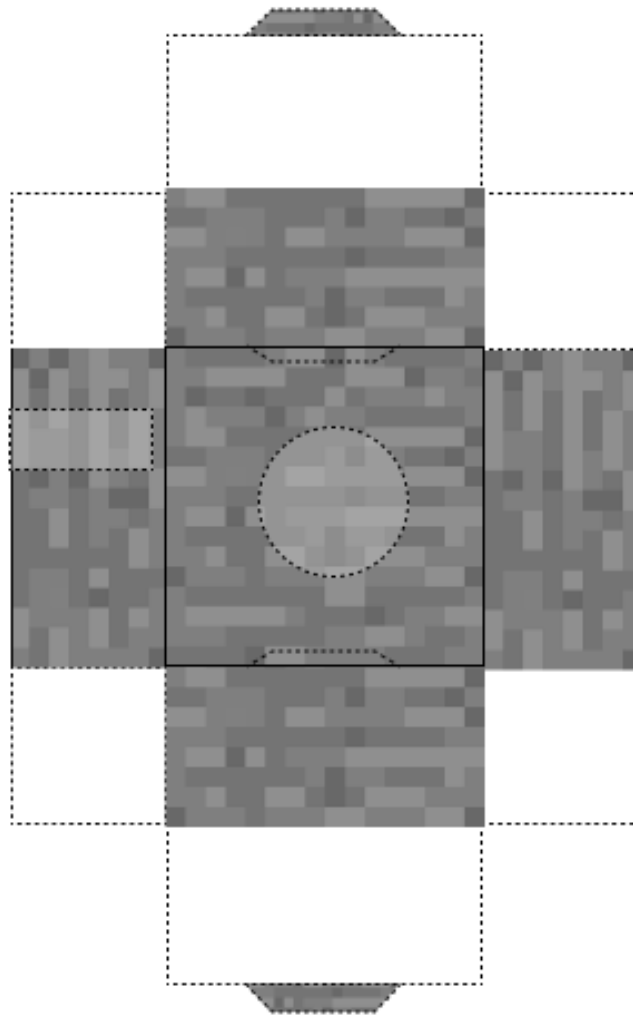


Εικόνα 7 Σύνδεση ενός κουμπιού στο GPIO χρησιμοποιώντας ένα breadboard και δύο καλώδια γεφύρωσης (Jumper).
Κατανοώντας τις διαφορετικές ακίδες GPIO, θα πρέπει να ρίξετε μια ματιά στην Εικόνα 8 που περιγράφει τις ιδιότητες κάθε ακίδας.

5V Power	5V Power	GND	GPIO14 UART0_TXD	GPIO15 UART0_RXD	GPIO18 PCM_CLK	GND	GPIO23	GPIO24	GND	GPIO25	GPIO8 SPI0_CEO_N	GPIO7 SPI0_CEL_N	ID_SD I ² C ID EEPROM	GND	GPIO12	GND	GPIO16	GPIO20	GPIO21
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
3V3 Power	GPIO2 SDA1 I ² C	GPIO3 SCL1 I ² C	GPIO4 1-Wire	GND	GPIO17	GPIO27	GPIO22	3V3 Power	GPIO10 SPI0_MOSI	GPIO9 SPI0_MISO	GPIO11 SPI0_SCLK	GND	ID_SD I ² C ID EEPROM	GPIO5	GPIO6	GPIO13	GPIO19	GPIO26	GND

Εικόνα 8 Raspberry Pi GPIO Ακίδες

Αφού δημιουργήσετε το κύκλωμα, πρέπει να συναρμολογήσετε το χάρτινο μπλοκ και να τοποθετήσετε το κύκλωμα σε αυτό. Για το σκοπό αυτό, πρέπει να χρησιμοποιήσετε την παρακάτω εκτυπώσιμη εικόνα. Για την καλύτερη χρήση, φροντίστε να εκτυπώσετε την κατασκευή σε λευκό χαρτόνι A4 και, στη συνέχεια, κόψτε τις καθορισμένες άκρες.



Εικόνα 9 Εκτυπώσιμη κατασκευή χαρτοκιβωτίου για physical computing block με κουμπί.

1.5.3 Προγραμματισμός με Python

Απλώς ανοίξτε το Thonny Python στην επιφάνεια εργασίας του Raspberry Pi και, στη συνέχεια, φορτώστε το σενάριο που ονομάζεται "volcano.py" που μπορείτε να βρείτε στην επιφάνεια εργασίας ή αντιγράψτε και επικολλήστε το σενάριο απευθείας στον μεταγλωττιστή Python της επιλογής σας. Εάν αποφασίσετε να συνδέσετε το κουμπί σε μια διαφορετική ακίδα GPIO, βεβαιωθείτε ότι έχετε αλλάξει τον αριθμό GPIO στο παρακάτω σενάριο (γραμμές 9 και 13).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minecraft
import mcpi.block as block
import random, time

mc = minecraft.Minecraft.create()

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

mc.postToChat('Press button for eruption!')
while True:
    if GPIO.input(23) == GPIO.HIGH: #Look for button press
        mc.postToChat("Eruption")
```



```
x, y, z = mc.player.getPos()
mc.setBlocks(x, y, z, x+20, y+15, z+20, block.AIR)

height = 10
center = x+15, y, z+15

for i in range(height): #Create the base
    size = height - i
    mc.setBlocks(center[0] - size, center[1] + i, center[2] - size, center[0] +
size, center[1] + i, center[2] + size, block.STONE)

while True:
    mc.setBlock(center[0], center[1]+height, center[2], block.LAVA_FLOWING)
    time.sleep(1)
```

1.6 Καύσωνας

1.6.1 Εξοπλισμός και σενάριο

Αυτή η δραστηριότητα θα παρουσιάσει ένα κύμα καύσωνα στον κόσμο του Minecraft Pi. Οι μαθητές θα συνδέσουν έναν αισθητήρα υπερήχων στο GPIO και θα τον προγραμματίσουν να δημιουργήσει ένα κύμα καύσωνα όταν ένα αντικείμενο πλησιάζει τον αισθητήρα.

Για αυτή τη δραστηριότητα θα χρησιμοποιήσετε τον παρακάτω εξοπλισμό:

1. 1 x Breadboard
2. 1 x HC-SR04 Αισθητήρας υπερήχων
3. 3 x 1k Ohm Αντιστάσεις
4. 4 x Καλώδια γεφύρωσης (Jumper) από αρσενικό σε θηλυκό
5. 1 x Κατασκευή χαρτοκιβωτίου

Αφού συνδεθούν όλα, οι μαθητές θα πρέπει να φορτώσουν τον χάρτη "Fishing Town" και το σενάριο Python με το όνομα "heatwave.py" που μπορεί να βρεθεί στην επιφάνεια εργασίας κάθε STEM4CLIM8 κονσόλας.

Καθώς ο «ήλιος» πλησιάζει στη γη, το φαινόμενο του καύσωνα θα εμφανιστεί προκαλώντας το θάνατο δέντρων, φυτών και χόρτων, παρόμοια με την πραγματική καταστροφή.

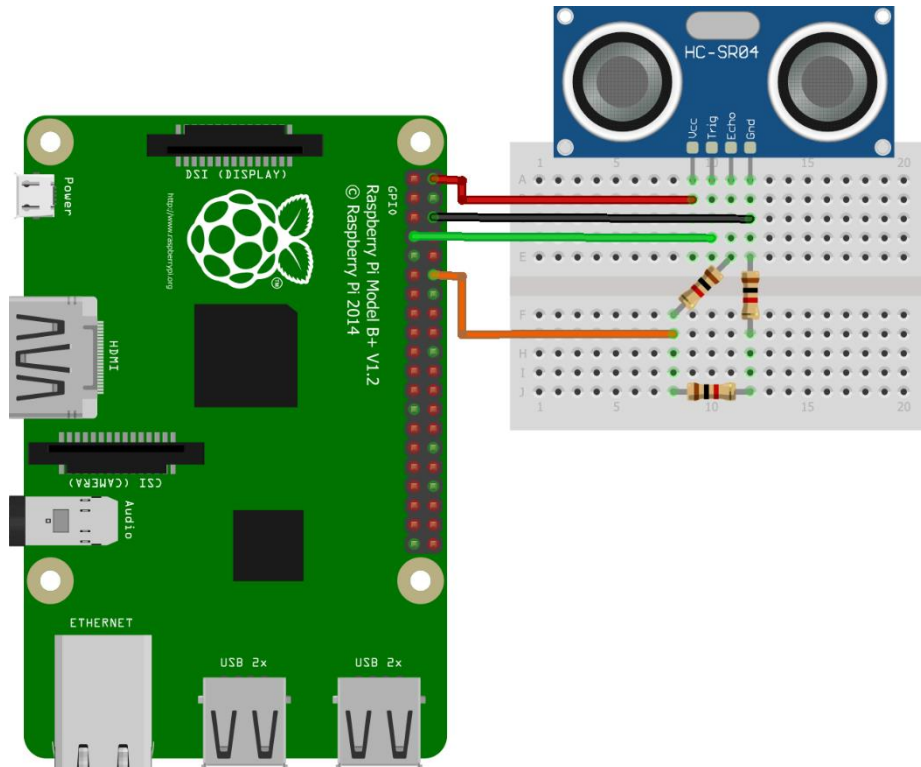
Μετά τη δραστηριότητα θα πρέπει να ακολουθήσει συζήτηση με τους μαθητές σας για το φαινόμενο της αύξησης της θερμοκρασίας και τα αρνητικά αποτελέσματα στο φυσικό περιβάλλον.

1.6.2 Ηλεκτρονικό κύκλωμα και physical block

Το πρώτο πράγμα που πρέπει να κάνετε είναι να δημιουργήσετε το κύκλωμα και να συνδέσετε τον αισθητήρα HC-SR04 στις ακίδες GPIO του Raspberry Pi. Πριν προχωρήσετε, πρέπει να απενεργοποιήσετε το Raspberry Pi και να το αποσυνδέσετε. Το πλήρες κύκλωμα είναι στο σχηματικό διάγραμμα που ακολουθεί.

Ο αισθητήρας απόστασης υπερήχων HC-SR04 διαθέτει 4 ακίδες: τροφοδοσία (VCC), σκανδάλη (TRIG), ηχώ (ECHO) και γείωση (GND). Η ακίδα τροφοδοσίας θα συνδεθεί με την ακίδα 5V του Raspberry Pi, η σκανδάλη θα εκχωρηθεί σε μια ακίδα GPIO ως έξοδος (ακίδα 4), η ηχώ θα εκχωρηθεί σε μια ακίδα GPIO ως είσοδος (ακίδα 18) και η γείωση θα συνδεθεί σε ακίδα γείωσης στο Raspberry Pi GPIO.

Το πρόγραμμα θα λειτουργεί με τέτοιο τρόπο ώστε κάθε φορά που ένα αντικείμενο πλησιάζει τον αισθητήρα, ο αισθητήρας εκπέμπει παλμό υπερήχων και το πρόγραμμα συνεχίζει να χρονομετρεί πόσο χρόνο χρειάζεται για να λάβει πίσω την ηχώ (δηλαδή πόσος χρόνος πέρασε όσο εκπέμπονταν τα ηχητικά κύματα, χτύπησαν το αντικείμενο μπροστά από τον αισθητήρα, αναπήδησαν και επέστρεψαν στον αισθητήρα).



fritzing

Εικόνα 10 Σχηματικό διάγραμμα κυκλώματος με αισθητήρα απόστασης συνδεδεμένο σε ακίδες GPIO.

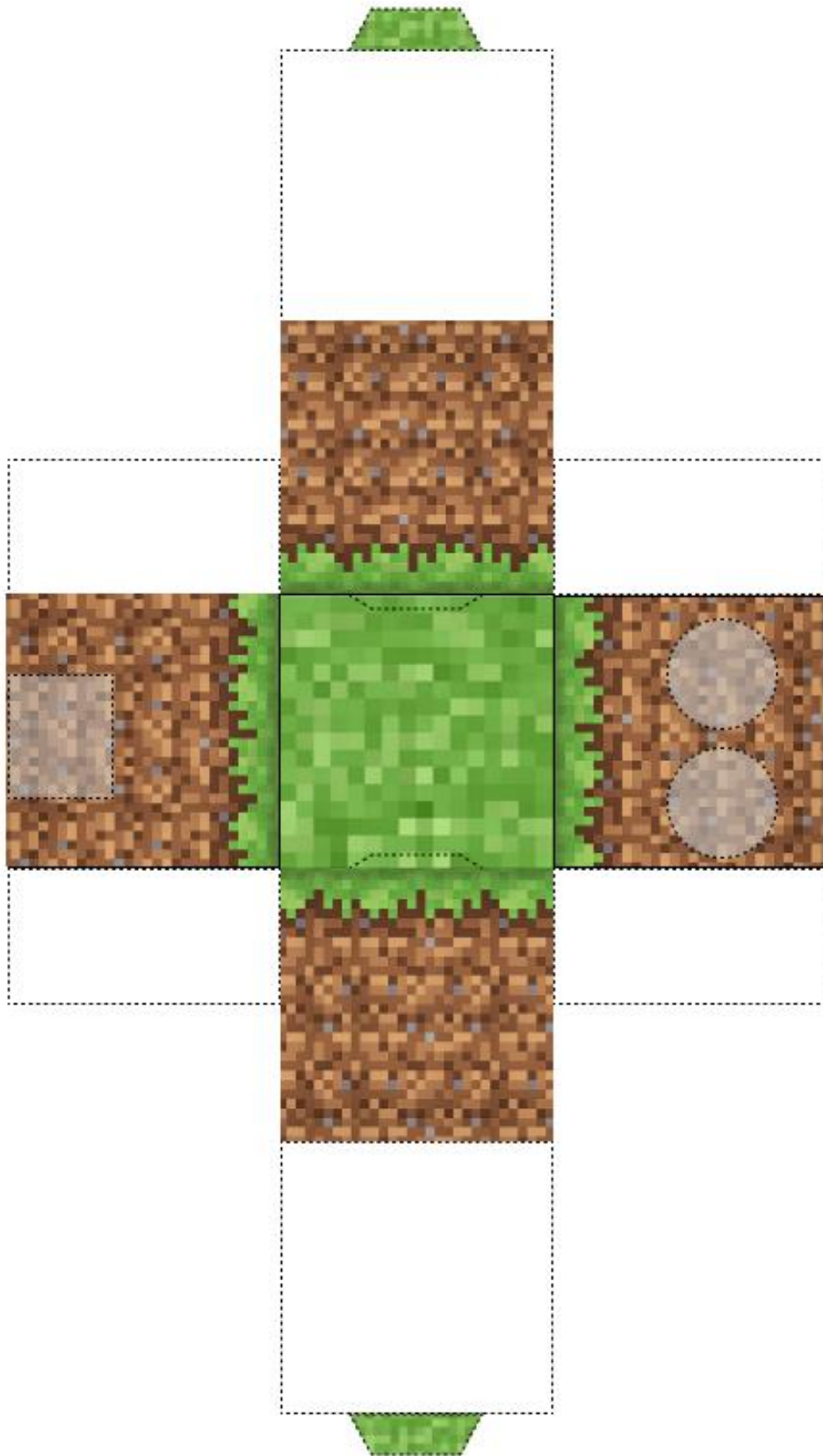
Κατανοώντας τις διαφορετικές ακίδες GPIO, θα πρέπει να δείτε την Εικόνα 11 που περιγράφει τις ιδιότητες κάθε ακίδας.

5V Power	5V Power	GND	GPIO14 UART0_TXD	GPIO15 UART0_RXD	GPIO18 PCM_CLK	GND	GPIO23	GPIO24	GND	GPIO25	GPIO8 SPI0_CEO_N	GPIO7 SPI0_CEL_N	ID_SD PCID EEPROM	GND	GPIO12	GND	GPIO16	GPIO20	GPIO21
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
3V3 Power	GPIO2 SDA1 I ² C	GPIO3 SCL1 I ² C	GPIO4 1-Wire	GND	GPIO17	GPIO27	GPIO22	3V3 Power	GPIO10 SPI0_MOSI	GPIO9 SPI0_MISO	GPIO11 SPI0_SCLK	GND	ID_SD PCID EEPROM	GPIO5	GPIO6	GPIO13	GPIO19	GPIO26	GND

Εικόνα 11 Raspberry Pi GPIO Ακίδες

Αφού δημιουργήσετε το κύκλωμα, πρέπει να συναρμολογήσετε μια κατασκευή από χαρτοκιβώτιο DIY και να τοποθετήσετε το κύκλωμα σε αυτό. Για το σκοπό αυτό, πρέπει να χρησιμοποιήσετε την παρακάτω εκτυπώσιμη εικόνα. Για καλύτερη εμπειρία, φροντίστε να

εκτυπώσετε την κατασκευή σε λευκό χαρτόνι A4 και μετά να κόψετε τις καθορισμένες άκρες.



Εικόνα 12 Εκτυπώσιμη χάρτινη κατασκευή για physical computing block με αισθητήρα απόστασης.

Η κατασκευή χαρτοκιβωτίων DIY αντιπροσωπεύει τη «Γη». Θα χρειαστείτε επίσης ένα άλλο αντικείμενο παρόμοιου μεγέθους που θα αντιπροσωπεύει τον «Ήλιο». Στη συσκευασία STEM4CLIM8, θα βρείτε μια τρισδιάστατη εκτυπωμένη σφαίρα που μπορείτε να χρησιμοποιήσετε. Επιπλέον, μπορείτε να εκτυπώσετε τρισδιάστατα τον δικό σας «ήλιο» χρησιμοποιώντας τα σχετικά αρχεία .stl που περιλαμβάνονται στην επιφάνεια εργασίας του Raspberry Pi.

1.6.3 Προγραμματισμός με Python

Όταν τελειώσετε με το κύκλωμα, μπορείτε να ενεργοποιήσετε το Raspberry Pi και να ξεκινήσετε το Thonny Python, στη συνέχεια να φορτώσετε το σενάριο που ονομάζεται "heatwave.py" το οποίο μπορείτε να βρείτε στην επιφάνεια εργασίας ή να αντιγράψετε και να επικολλήσετε το σενάριο απευθείας στον μεταγλωττιστή Python της επιλογής σας. Εν ολίγοις, όταν εκτελείτε το πρόγραμμα, ο αισθητήρας απόστασης HC-SR04 θα αρχίσει να εκπέμπει ριπή υπερήχων. Κάθε φορά που πλησιάζετε τον «ήλιο» στον αισθητήρα («Γη») το φαινόμενο του καύσωνα θα συμβαίνει στον κόσμο του Minecraft Pi. Θα δείτε δέντρα, λουλούδια και φυτά να παίρνουν φωτιά και να γίνονται στάχτη.

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minecraft
import mcpi.block as block
mc = minecraft.Minecraft.create()
import random, time

GPIO.setwarnings(False)
GPIO.cleanup()
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) #pin for push button

#defining TRIG and ECHO pins
TRIG = 4
ECHO = 18
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)

def heatwave(x, z):
    mc.postToChat('Heatwave!')
    y = mc.getHeight(x, z)
    endtime = time.time() + random.randint(50, 90)
    while time.time() < endtime:
        blkid = block.AIR.id
        while blkid == block.AIR.id:
            bx = random.randint(x-10, x+10)
            by = random.randint(y, y+10)
            bz = random.randint(z-10, z+10)
            blkid = mc.getBlockWithData(bx, by, bz).id
        blk = blkid
        blkd = mc.getBlockWithData(bx, by, bz).data
        if blkid == block.GRASS.id:
            blk = block.DIRT.id
            blkd = 0
        elif blkid in [block.WATER.id, block.WATER_FLOWING.id, block.WATER_STATIONARY.id]:
            blk = block.WATER.id
            blkd = 1
        elif blkid == block.LEAVES.id:
            blk = block.COBWEB.id
            blkd = 0
        elif blkid == block.WOOD.id:
            blk = block.LAVA_STATIONARY.id
            blkd = 1
        mc.setBlock(bx, by, bz, blk, blkd)
```

```
disasters = [heatwave]
def main(disasters, mc):
    based = random.randint(1, 10000)
    mc.postToChat('Push button for sensor to work!')
    mc.postToChat('Move "sun" closer to sensor, and wait for Heatwave!')
    #emit a burst of ultrasound
    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)
    StartTime = time.time()
    StopTime = time.time()
    while True:
        if GPIO.input(23) == GPIO.HIGH:
            while GPIO.input(ECHO) == False:
                StartTime = time.time()
            while GPIO.input(ECHO) == True:
                StopTime = time.time()
            TimeElapsed = StopTime - StartTime
            distance = (TimeElapsed * 34300) / 2
            while distance < 15:
                print (distance)
                t = random.randint(5, 120)
                t = 5
                time.sleep(t)
                random.seed(based + t)
                based = random.randint(1, 10000)
                random.shuffle(disasters)
                disaster = random.choice(disasters)
                ppos = mc.player.getTilePos()
                disaster(ppos.x, ppos.z)
try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

1.7 Πλημμύρα

1.7.1 Εξοπλισμός και σενάριο

Αυτή η δραστηριότητα θα παρουσιάσει μια πλημμύρα στον κόσμο του Minecraft Pi. Οι μαθητές θα συνδέσουν έναν αισθητήρα σταγόνας βροχής στο GPIO και θα τον προγραμματίσουν να δημιουργεί πλημμύρα κάθε φορά που ο αισθητήρας αγγίζει το νερό.

Η ιδέα είναι να δημιουργήσετε ένα physical computing block χρησιμοποιώντας τον ακόλουθο εξοπλισμό:

1. 1 x Breadboard
2. 1 x Αισθητήρας ανίχνευσης σταγόνων βροχής
3. 3 x Καλώδια γεφύρωσης (Jumper) από αρσενικό σε θηλυκό
4. 2 x Καλώδια γεφύρωσης (Jumper) από θηλυκό σε θηλυκό
5. 1 x Κατασκευή χαρτοκιβωτίου

Αφού συνδεθούν όλα, οι μαθητές θα πρέπει να φορτώσουν τον χάρτη "Fishing Town" και το σενάριο Python με το όνομα "flood.py" που μπορεί να βρεθεί στην επιφάνεια εργασίας κάθε κονσόλας STEM4CLIM8.

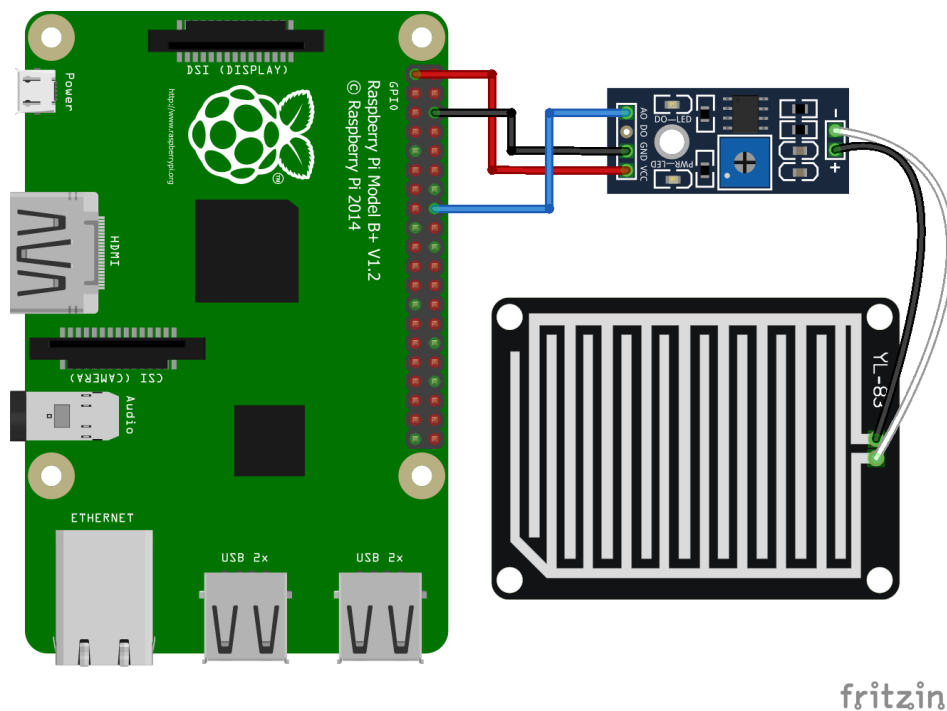
Στη συνέχεια, κάθε φορά που πέφτει νερό στον αισθητήρα, θα πρέπει να συμβαίνει μια πλημμύρα στον κόσμο του Minecraft Pi προκαλώντας τον πνιγμό των πάντων. Η έντονη βροχόπτωση θα αναφέρεται με ένα μήνυμα στον κόσμο του Minecraft που θα δημιουργήσει ένα Geyser (πλημμύρα).

Μετά τη δραστηριότητα θα πρέπει να ακολουθήσει μια συζήτηση με τους μαθητές σας σχετικά με τις λάθος πρακτικές και τους λόγους που σημειώθηκε η πλημμύρα μετά τη δυνατή βροχή.

1.7.2 Ηλεκτρονικό κύκλωμα και *physical block*

Η σύνδεση ενός αισθητήρα σταγόνας βροχής στο GPIO του Raspberry Pi είναι μια σχετικά εύκολη διαδικασία που φαίνεται στην Εικόνα 13 παρακάτω:

1. Η πλακέτα ελέγχου του αισθητήρα σταγόνας βροχής έχει 4 ακίδες που μπορούν να συνδεθούν σε GPIO, συγκεκριμένα, τάση (VCC), γείωση (GND), ψηφιακή έξοδο (DO) και αναλογική έξοδο (AO).
2. Πρέπει να συνδέσετε τα παρακάτω μέσω ενός breadboard ή απευθείας στο GPIO:
 - a. VCC σε 5V (ακίδα 2)
 - b. GND σε GND (ακίδα 4)
 - c. DO σε GPIO23 (ακίδα 16)
3. Στη συνέχεια, πρέπει να συνδέσετε την πλακέτα ανίχνευσης αισθητήρα στην πλακέτα ελέγχου χρησιμοποιώντας δύο καλώδια γείωσης (jumper), συνδέοντας τα ακόλουθα:
 - a. Η πλευρά + της πλακέτας ελέγχου στην πλευρά + της πλακέτας ανίχνευσης.
 - b. Η - πλευρά του πίνακα ελέγχου προς την - πλευρά του πίνακα ανίχνευσης.
4. Το κύκλωμά σας είναι έτοιμο.



fritzing

Εικόνα 13 Σχηματική απεικόνιση του αισθητήρα σταγόνας βροχής που συνδέεται με τις ακίδες GPIO του Raspberry Pi.

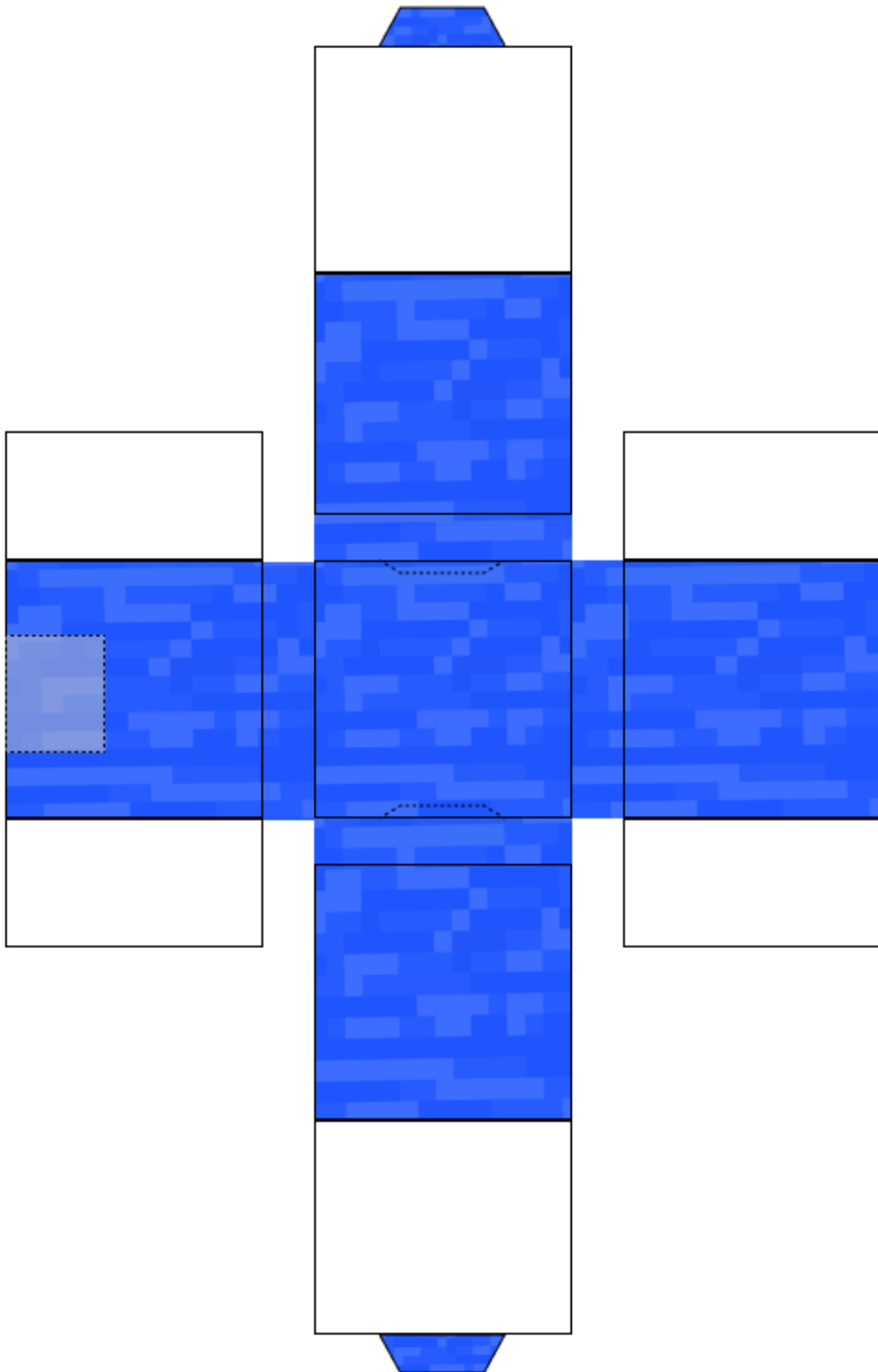
Κατανοώντας τις διαφορετικές ακίδες GPIO, θα πρέπει να δείτε την Εικόνα 14 που περιγράφει τις ιδιότητες κάθε ακίδας.



Εικόνα 14 Raspberry Pi GPIO ακίδες

Ένας αισθητήρας σταγόνων βροχής μπορεί να στείλει αναλογικές και ψηφιακές εξόδους στη συσκευή ελέγχου. Σε αυτήν την περίπτωση, ο υπολογιστής Raspberry Pi διαβάζει μόνο ψηφιακά σήματα. Έτσι, για αυτή τη δραστηριότητα, χρειάζεται μόνο ψηφιακή έξοδος, καθώς το πρόγραμμα θα εκτελείται κάθε φορά που ο αισθητήρας εντοπίζει μια σταγόνα βροχής, προκαλώντας πλημμύρα στον κόσμο του Minecraft Pi.

Αφού δημιουργήσετε το κύκλωμα, πρέπει να συναρμολογήσετε μια κατασκευή από χαρτοκιβώτιο DIY και να τοποθετήσετε το κύκλωμα σε αυτό. Για το σκοπό αυτό, πρέπει να χρησιμοποιήσετε την παρακάτω εκτυπώσιμη εικόνα. Για καλύτερη εμπειρία, φροντίστε να εκτυπώσετε την κατασκευή σε λευκό χαρτόνι A4 και, στη συνέχεια, κόψτε τις καθορισμένες άκρες.



Εικόνα 15 Εκτυπώσιμη κατασκευή χαρτοκιβωτίου για physical computing block με αισθητήρα σταγόνας βροχής.

1.7.3 Προγραμματισμός με Python

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN)

def geyser(x, z):
    mc.postToChat('Flood!')
    y = mc.getHeight(x, z)
    mc.setBlocks(x-2, y+5, z-2, x+2, -60, z+2, block.WATER.id)
    time.sleep(25)
    mc.setBlocks(x-2, y+5, z-2, x+2, -60, z+2, block.AIR.id)

disasters = [geyser]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Drop water on sensor for disaster!')
    while True:
        if GPIO.input(23) == GPIO.LOW: #Look for raindrop sensor activation
            t = random.randint(5, 120)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

1.7 Συμπέρασμα

Η χρήση αυτών των δραστηριοτήτων θα είναι για τα παιδιά ώστε να βιώσουν τις φυσικές καταστροφές και να κατανοήσουν τον ρόλο των ανθρώπινων δραστηριοτήτων στην επιδείνωση αυτών των φαινομένων, όπως η κλιματική αλλαγή κ.λπ.

References

Minecraft Pi Edition: SURVIVAL mode-ish. Διαθέσιμο στο: <https://teachwithict.weebly.com/computing-blog/minecraft-pi-edition-survival-mode-ish#sthash.S6XeUUDH.dpbs> Τελευταία πρόσβαση: 19/11/2021

How to use Rain Sensor Module with Arduino & Raspberry Pi by SunFounder Maker Education. Διαθέσιμο στο: <https://www.youtube.com/watch?v=Xnisf0GP9bA&t=203s> Τελευταία πρόσβαση: 19/11/2021

Using a Raspberry Pi distance sensor (ultrasonic sensor HC-SR04). Διαθέσιμο στο: <https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/> Τελευταία πρόσβαση: 19/11/2021