

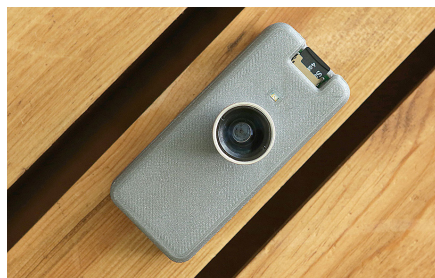
# Przenośny rejestrator obrazów z Raspberry Pi Zero

Wybieracie się na wakacje i chcielibyście zachować jak najwięcej wspomnień z nich w postaci obrazów, ale nie przepadacie za ciągłym sięganiem po aparat fotograficzny? Mamy dla Was rewelacyjny projekt, który pozwoli w dosyć szybki sposób przygotować niedrogi, estetyczny, automatyczny aparat fotograficzny, np. do powieszenia na szyi. Omawiany projekt bazuje na Raspberry Pi Zero i korzysta z druku 3D, ale sprawdzi się także z innymi wersjami RPI.

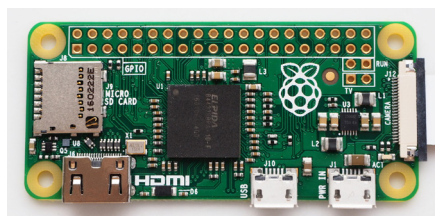
Większość projektów, które opisujemy na łamach Elektroniki Praktycznej przypomina raczej prototypy niż gotowe urządzenia. To dlatego, że ich twórcy koncentrują się najczęściej na elektronice, a obudowa jest albo prowizoryczna, albo przede wszystkim zabawna. Raspberry Pi Wearable Time Lapse Camera, opracowana przez Phillipa Burgessa i braci Ruiz z USA mocno wyróżnia się na tym tle, gdyż w praktyce ma cechy gotowego, dopracowanego urządzenia i całkiem atrakcyjny wygląd. A dzięki dostępności wszystkich podzespołów i materiałów można szybko samodzielnie odtworzyć projekt i zabrać go na najbliższe wakacje. Co więcej, to jeden z nielicznych w EP projektów urządzenia noszonego, a więc tzw. typu „wearable”. Jest to również okazja by wykorzystać nowe Raspberry Pi Zero, które po wyczerpaniu poprzedniej wersji jest teraz dostępne w sprzedaży.

## Idea

Twórcy Raspberry Pi Wearable Time Lapse Camera postawili sobie bardzo jasny cel – chodziło im o stworzenie wygodnego



Rysunek 1. Zmontowany rejestrator, gotowy do pracy



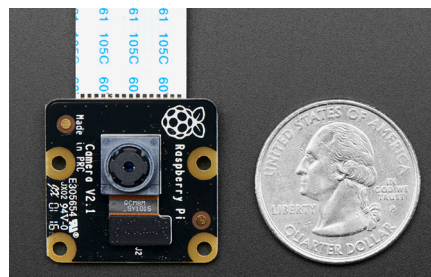
Rysunek 2. Raspberry Pi Zero v1.3

w użyciu, przenośnego rejestratora obrazów. Tak by można było robić np. prezentacje typu timelapse. Nic więcej. Nie starali się ani uzyskać nie wiadomo jakiej wydajności, możliwości nagrywania wideo ani wyprowadzać wielu przycisków do konfiguracji. Stąd mogli skoncentrować się na tym, co było dla nich najważniejsze i względnie szybko przygotować cały projekt.

W rezultacie rejestrator działa w dosyć prosty sposób. Jego domyślne ustawienia sprawiają, że po podłączeniu zasilania i uruchomieniu systemu operacyjnego kamera zaczyna wykonywać fotografie z odstępami co 15 sekund. Obrazy zapisywane są w jednym konkretnym katalogu i zawierają znaczki czasowe w nazwach. Po wyłączeniu rejestratora można wyjąć kartę pamięci i włożyć ją do czytnika, po czym samodzielnie zgrać zdjęcia. Niby proste, ale całość wykonania zdecydowanie zasługuje na uwagę. Jak się sprawdza w praktyce? Można zobaczyć na wideo pod adresem <https://goo.gl/maqaAc>.

## Elektronika

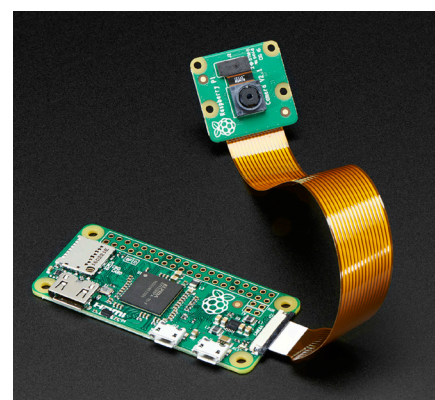
Podstawowe komponenty użyte w projekcie to Raspberry Pi Model Zero w wersji 1.3 (rysunek 2), Raspberry Pi Camera w 8-megapikselowej wersji V2 (rysunek 3) oraz polimerowy akumulator litowo-jonowy o pojemności 500 mAh (rysunek 4). Zarówno Raspberry Pi, jak i moduł kamery można swobodnie zastąpić inną wersją – choćby klasycznym, dużym RPI, a kamerę starszym modelem, bądź nawet wersją na podczerwień.



Rysunek 3. Moduł kamery do Raspberry Pi



Rysunek 4. Akumulator użyty w projekcie



Rysunek 5. Połączenie Raspberry Pi Zero z kamerą

Użyte podzespoły i napisane oprogramowanie są na tyle uniwersalne, że będą działać z każdym z tych komponentów.

Moduł kamery został podłączony do Raspberry Pi za pomocą adekwatnej taśmy (rysunek 5), takiej jak dostępna pod adresem <https://goo.gl/D9Zgr7>. Użyto karty pamięci 4 GB, ale jeśli chce się zapisywać bardzo dużo zdjęć, warto sięgnąć po większą pamięć. Sekcję zasilania wzbogacono o moduł ładowarki (rysunek 6), który umożliwia jednocześnie ładowanie akumulatora i zasilanie komputera. Jest on dosyć drogi, gdyż kosztuje 15 dolarów

REKLAMA

Projekty na <sup>000</sup>

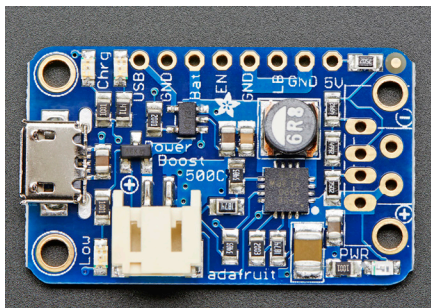
# STM32

[www.stm32.eu](http://www.stm32.eu)

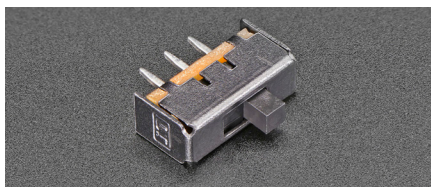
**ST** life.augmented

**KAMAMI**

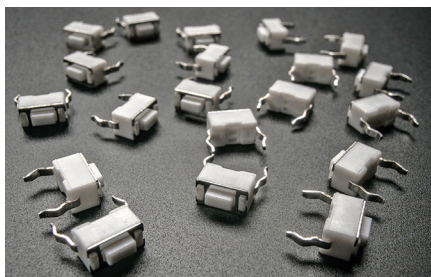
(<https://goo.gl/b4o27h>), stąd znacznie podnosi cenę projektu. Zamiast tego można po prostu bezpośrednio podłączyć napięcie 5 V z zasilacza, lub użyć prostszej ładowarki. Pozostałe komponenty to przełącznik suwakowy (rysunek 7), przełącznik przyciskowy monostabilny (rysunek 8) i dioda LED



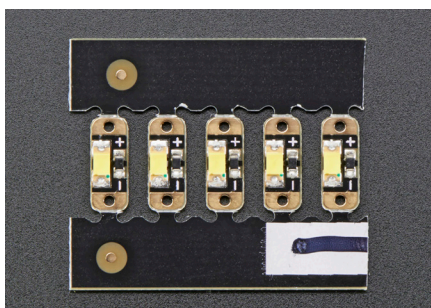
Rysunek 6. Moduł ładowania akumulatora Adafruit PowerBoost 500 Charger



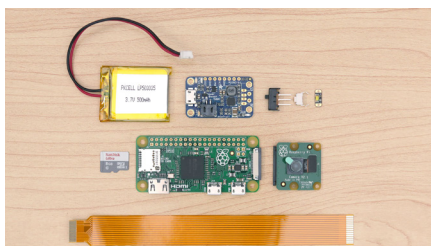
Rysunek 7. Przełącznik bistabilny, przesuwany



Rysunek 8. Przełączniki monostabilne, takie jak użyte w projekcie



Rysunek 9. Moduły diod LED takie, jak użyte w projekcie



Rysunek 10. Wszystkie komponenty elektroniczne rejestratora

z rezystorem 100 Ω w postaci gotowego modułiku (rysunek 9). Wszystkie komponenty zebrano na rysunku 10.

Połączenia pomiędzy podzespołami są bardzo proste. Kamerę należy podłączyć tasiemką do złącza na Raspberry Pi. Układ zasilania (wyprowadzenia + i -) należy podłączyć do pinów zasilania 5 V i masy w Raspberry Pi. Diodę LED z szeregowo podłączonym rezystorem należy wpiąć pomiędzy piny GPIO5 i masę. Przycisk monostabilny pomiędzy piny GPIO21 i masę. Posłuży do bezpiecznego wyłączenia rejestratora. Wyprowadzenie akumulatora trzeba podpiąć do jedyne pasującego złącza na płytce PowerBoost. Przełącznik monostabilny podłączamy pomiędzy wejście EN a masę układu PowerBoost, dzięki czemu będzie służyć jako włącznik całego rejestratora. Układ wyprowadzeń Raspberry Pi Zero pokazano na rysunku 11.

### Mechanika

Wielką zaletą opisywanego projektu jest dobrze dopasowana obudowa mechaniczna. Składa się ona z trzech podstawowych części, które można wydrukować nawet na niewielkiej drukarce 3D (rysunek 12). Wierzchnią i spodnią część obudowy uzupełniają klipsy, który pozwala zamontować kamerę np. na pasku. Ma on też otwór, dzięki któremu cały rejestrator można zawiesić na cienkiej lince.

Projekt obudowy można bezpłatnie pobrać z repozytorium Thingiverse (<https://goo.gl/Ur8jNp>). Wizualizację modelu pokazano na rysunku 13. Wśród plików 3D jest także fragment do zamontowania statywu. Pliki są w formacie STL, a więc osoby umiające projektować mogą sobie swobodnie zmodyfikować obudowę tak, by pasowała do innych podzespołów, w tym np. do większego akumulatora. Ci czytelnicy, którzy nie mają takich zdolności, a chcieliby użyć nieco innych podzespołów będą w gorszej sytuacji, ale w praktyce większość modyfikacji będzie polegała na rezygnacji np. z modułu PowerBoost, po którym wolną przestrzeń można po prostu zapchać klejem termicznym. Zresztą twórcy i tak wykorzystali klej w swoim projekcie.

Sama obudowa została zaprojektowana tak, by idealnie pasować do użytych komponentów. Najlepiej wydrukować ją z użyciem poliaktydu (PLA). Twórcy zalecają delikatnie zwiększyć model o jakieś 2%, jeśli ma być wykonywany z filamentu ABS.

### Montaż

Dobrze dopasowana obudowa i precyzyjnie dobrane komponenty sprawiają, że aby uzyskać idealne spasowanie części, trzeba je odpowiednio zmontować.

Przełącznik przesuwany należy pozbyć jednego z bocznych wyprowadzeń,

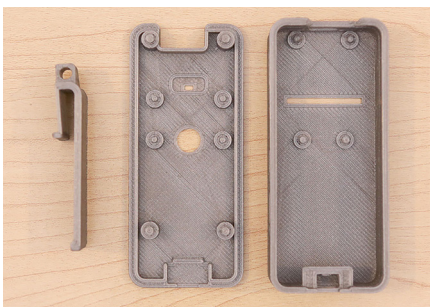
Pi Model B/B+			
3V3 Power	1	2	5V Power
GPIO2 SDA1 I2C	3	4	5V Power
GPIO3 SCL1 I2C	5	6	Ground
GPIO4	7	8	GPIO14 UART0_TXD
Ground	9	10	GPIO15 UART0_RXD
GPIO17	11	12	GPIO18 PCM_CLK
GPIO27	13	14	Ground
GPIO22	15	16	GPIO23
3V3 Power	17	18	GPIO24
GPIO10 SPI0_MOSI	19	20	Ground
GPIO9 SPI0_MISO	21	22	GPIO25
GPIO11 SPI0_SCLK	23	24	GPIO8 SPI0_CE0_N
Ground	25	26	GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM	27	28	ID_SC I2C ID EEPROM
GPIO5	29	30	Ground
GPIO6	31	32	GPIO12
GPIO13	33	34	Ground
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
Ground	39	40	GPIO21

Rysunek 11. Rozkład wyprowadzeń Raspberry Pi Zero

a pozostałe dwa przylutować do cienkich przewodów. Moduł LED należy również podłączyć do dwóch przewodów, a jeśli stosujemy oddzielną diodę zamiast modułu, najlepiej rezystor dolutować tak by rezystor można było swobodnie ułożyć w obudowie. Przycisk monostabilny też lutujemy do dwóch cienkich przewodów. Przewody od przełącznika bistabilnego montujemy do przelotek EN i GND na module ładowarki (rysunek 14). Wyprowadzenia ładowarki łączymy przewodami z Raspberry Pi Zero – najlepiej od spodu. Następnie, również od spodu lutujemy wyprowadzenia diody LED z opornikiem. Kolejny krok

to podłączenie taśmą kamery z komputerem i podłączenie akumulatora w gniazdo na module zasilania. Przycisku monostabilnego nie należy na tym etapie trwale lutować do Raspberry Pi ze względu na konstrukcję obudowy. Całość po wykonaniu połączeń elektrycznych powinna wyglądać tak, jak na **rysunku 15**.

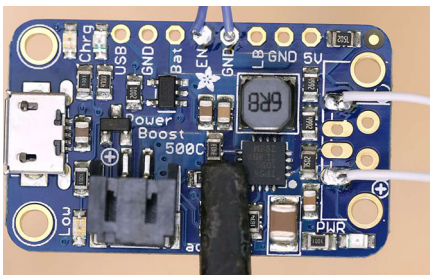
W praktyce można już sprawdzić czy wszystko poprawnie działa (choć wypada mieć wcześniej przygotowany system, o którym napiszemy w kolejnym kroku). Podpięcie wszystkich komponentów wraz z naładowanym akumulatorem może spowodować uruchomienie Raspberry Pi, o ile tylko przełącznik przesuwany jest w odpowiedniej pozycji. Włączenie przetwornicy jest sygnalizowane niebieską diodą. Raspberry Pi uruchamia się około minuty przy pierwszym



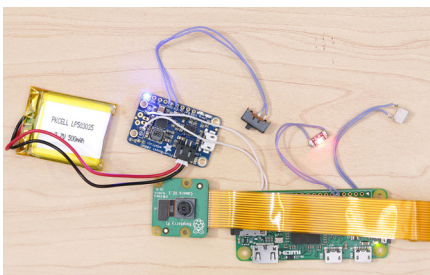
**Rysunek 12. Główne komponenty obudowy**



**Rysunek 13. Model 3D obudowy**



**Rysunek 14. Sposób przyłutowania przewodów do modułu ładowarki**



**Rysunek 15. Połączone ze sobą komponenty elektroniczne**

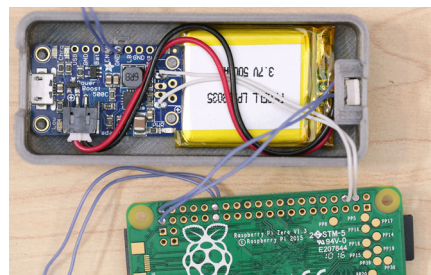
starciu i mniej więcej połowę krócej przy kolejnych restartach. Czas ten może się wydłużać, jeśli na karcie pamięci znajdzie się bardzo wiele plików graficznych. W domyślnym trybie pracy, za każdym razem gdy wykonywana jest fotografia, zapala się też dioda LED przyłutowana do GPIO5.

Aby bezpiecznie wyłączyć rejestrator trzeba przez kilka sekund przetrzymać przycisk podłączony do GPIO21, który powoduje wysłanie polecenia HALT do systemu i wstrzymanie jego pracy. Po zakończeniu pracy urządzenia można wyjąć kartę i sprawdzić, czy zdjęcia zostały wykonane.

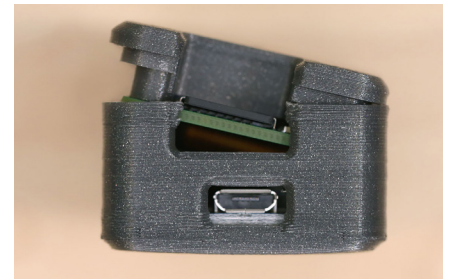
Zlutowane ze sobą elementy można umieszczać w obudowie, ale najlepiej zacząć od przycisku monostabilnego, którego wyprowadzenia należy przeciągnąć przez otwór w jej spodzie i następnie już na stałe przyłutować do Raspberry Pi. W drugiej kolejności wypada zamontować przetwornicę (tak by weszła na wypustki w obudowie) i umieścić obok niej akumulator. Jeśli elementy obudowy nie są idealnie spasowane po druku 3D, można użyć kleju by podzępoły nie poruszały się luźno. Diodę LED należy przymontować do otworu we froncie obudowy. Jeśli użyto innej niż w oryginalnym projekcie, prawdopodobnie będzie konieczne jej przyklejenie. Ważny krok to odpowiednie wypozycjonowanie modułu kamery. Tu również projektanci obudowy przygotowali odpowiednie wypustki, które pozwalają na sztywne uchwycenie kamery (**rysunek 16**).

W końcu można zamontować Raspberry Pi, uważając by odpowiednio zawiązać taśmę łączącą RPI z kamerą. Dobrym pomysłem jest wykorzystanie niewielkiej ilości taśmy izolacyjnej lub innego elastycznego, ale i lekkiego materiału, który zapobiegnie rozwijaniu się taśmki. Gdy niemal wszystko jest na miejscu można zamontować z boku przełącznik bistabilny. Wtedy można też zamknąć obudowę, która powinna w tym momencie wydać z siebie przyjemny „klik” (**rysunek 17**). Jeśli się to nie udaje, można spróbować umieścić przełącznik bistabilny w drugiej części obudowy i wtedy ponownie spróbować zamknąć całość.

Montaż klipsu wykonuje się na koniec poprzez wciśnięcie go w otwór z tyłu obudowy. Może się zdarzyć, że będzie on wymagał



**Rysunek 16. Sposób montażu komponentów w obudowie**



**Rysunek 17. Obudowa na chwilę przed zamknięciem**

użycia kleju, by pewnie trzymał się w miejscu (**rysunek 18**).

## Oprogramowanie

Twórcy przygotowali gotowy obraz ze skonfigurowanym systemem, który można pobrać z adresu: <https://goo.gl/Yfp4Vy> i wgrać go na kartę microSD o pojemności 4 GB. Plik ten jest spakowany i zajmuje 365 MB. Skorzystanie z tego pliku sprawia, że nie trzeba robić nic więcej. My jednak pokażemy, jak projekt zrealizować samodzielnie, a przy okazji wyjaśnimy na jakiej zasadzie działa.

Po pierwsze należy pobrać aktualną wersję Raspbiana – najlepiej w wersji Lite. Jest dostępny ze strony: <https://goo.gl/Zfo4ZG> i zgrać go na kartę pamięci. Przy pierwszym uruchomieniu trzeba wykonać polecenie `sudo raspi-config` i rozszerzyć partycję (opcja Expand Filesystem), włączyć kamerę (Enable Camera), a najlepiej też włączyć SSH i skonfigurować ustawienia klawiatury, by nie były brytyjskie. Należy się też upewnić, że komputer ma łączność z Internetem.

Po restarcie należy zainstalować oprogramowanie, które pozwoli na korzystanie z interfejsu GPIO. W tym celu używamy WiringPi:

```
sudo apt-get update
sudo apt-get install wiringpi
```

W katalogu /boot należy stworzyć plik timelapse.sh. Można to uczynić poleceniem `sudo nano /boot/timelapse.sh`.

Alternatywnie plik ten można wgrać bezpośrednio na kartę pamięci za pomocą czytnika, gdyż katalog /boot jest

REKLAMA



[www.stm32.eu](http://www.stm32.eu)

ST  
life.augmented

KAMAMI

widoczny na zwykłych komputerach z systemem Windows.

Plik `timelapse.sh` zawiera tak naprawdę praktycznie cały skrypt wraz z ustawieniami. Jego treść publikujemy na **listingu 1**.

Pozostaje tylko zapewnić, by skrypt uruchamiał się automatycznie po starcie Raspberry Pi. W tym celu w pliku `/etc/rc.local` należy dodać treść `sh /boot/timelapse.sh 2>/dev/null &`.

### Zasada działania

Wypada jeszcze wyjaśnić jak działa i jak zmienić ustawienia projektu. W praktyce Raspberry Pi Wearable Time Lapse Camera po prostu w pętli pobiera aktualny czas, sprawdza czy minęło przynajmniej 15 sekund (lub inny okres wpisany w konfiguracji) od wykonania poprzedniego zdjęcia, po czym uruchamia program do obsługi kamery i wykonuje zdjęcie. Jest ono zapisywane z kolejnymi numerami w katalogu docelowym. W pliku konfiguracyjnym oprócz interwału czasowego podana jest także rozdzielczość zdjęć oraz jakość kompresji. Sam program `raspistill` jest wywoływany z opcją `-t 250`, która mówi, że zdjęcie ma być wykonane 250 ms od uruchomienia programu. Może się zdarzyć, że do tego czasu kamera się nie ustabilizuje odpowiednio i w niektórych przypadkach konieczne jest zwiększenie tego czasu. Gdyby była potrzeba można też ręcznie za pomocą dalszych parametrów ustalić inne parametry ekspozycji, czy czułość sensora obrazu. Warto zaznajomić się z dokumentacją ze strony <https://goo.gl/eZM8S>.

Twórcy dodają także, że jeśli bardzo zależy nam na tym by nie stracić żadnego ujęcia, można dodać polecenie `sync` zaraz



Rysunek 18. Rejestrator z klipsem

po poleceniu wykonania fotografii. Spowoduje to, że na karcie pamięci poprawnie utrwalone zostaną ewentualne pliki przechowywane jeszcze w pamięci, tyle że proces ten trwa kilka sekund i znacznie wydłuża minimalny interwał pomiędzy kolejnymi ujęciami (z ok. 2 sekund do mniej więcej 10 sekund).

### Podsumowanie i ocena projektu

Zaprezentowany projekt jest nie tylko estetyczny, ale i praktyczny – może być faktycznie użyteczny. Stanowią też dobrą okazję do wypróbowania możliwości druku 3D. Opcjonalnie rejestrator można doposażyć w zewnętrzny obiektyw, taki jak do telefonów komórkowych i mocowany magnetycznie (**rysunek 19**). Całość projektu można obejrzeć pod adresem <https://goo.gl/QSiE5W>.

Od strony programistycznej jest bardzo prosty, choć bardzo podobny efekt dałoby się uzyskać w prostszy sposób. Mianowicie sam program `raspistill` pozwala na wykonywanie zdjęć z określonym interwałem czasowym – nawet krótszym niż te 2 sekundy, poniżej których trudno zejść w skrypcie przygotowanym na potrzeby projektu. Niestety `timelapse` wykonany bezpośrednio z użyciem ustawień `raspistill` będzie utrzymywać cały czas włączoną kamerę, która pobiera



Rysunek 19. Zamocowany na stojaku rejestrator z dodatkowym obiektywem

prąd z akumulatora. Do tego nowe moduły kamer nie mają diody LED, więc nie byłoby możliwości sygnalizowania momentów wykonywania zdjęć.

Warto jednak zauważyć, że pomimo starań o zmniejszenie poboru energii, twórcy wcale nie wykazali się w tym zakresie bardzo dużym wysiłkiem. Pętla nieskończona, w której pobierany jest aktualny czas i porównywany z poprzednim czasem fotografii wykonuje się niemal non stop, obciążając tym samym procesor komputera. Co prawda nie angażuje to obwodów graficznych ani niemal żadnych innych, ale znacznie bardziej eleganckim rozwiązaniem byłoby wykorzystanie bardziej zaawansowanego języka programowania (choćby pythona) i rozpisanie funkcji na wątki, które w trakcie oczekiwania na wykonanie praktycznie nie korzystałyby z procesora. Nic nie stoi na przeszkodzie, by taką modyfikację projektu wykonać już samodzielnie. W końcu wszystkie komponenty, wraz z projektem obudowy – nie ulegną zmianie.

Marcin Karbowniczek, EP

```
Listing 1. Kod skryptu rejestratora
#!/bin/sh
# konfiguracja
INTERVAL=15          # Czas pomiędzy pomiarami, wyrażony w sekundach
WIDTH=1280           # Szerokość zapisywanych obrazów
HEIGHT=720           # Wysokość zapisywanych obrazów
QUALITY=51           # Jakość obrazów (kompresji) JPEG, od 0 do 100
DEST=/boot/timelapse # Katalog w którym zapisywane są zdjęcia (nie może zawierać cyfr) PREFIX=img
                    # Prefiks nazw plików (nie może zawierać cyfr)
HALT=21              # Pin przycisku monostabilnego
LED=5                 # Pin diody LED
prevtime=0           # Resetowanie czasu wykonania ostatniego zdjęcia
gpio -g mode $HALT up # Inicjalizacja GPIO przycisku monostabilnego
gpio -g mode $LED out # Inicjalizacja GPIO diody stanu
mkdir -p $DEST       # Utworzenie katalogu docelowego, jeśli nie istnieje

# Poszukiwanie ostatniego z zapisanych zdjęć i zwiększenie jego numeru o 1
FRAME=$((find $DEST -name "*.jpg" -printf %f\n | sed ,s/^[^0-9]*/g/ | sort -zn | head -1 | sed ,s/^[^0-9]*/g/ + 1))
while :
do
while : # Pętla sprawdzająca czy minął odpowiedni czas
do
currenttime=$(date +%s) #pomiar aktualnego czasu
if [ $(($currenttime-$prevtime)) -ge $INTERVAL ]; then #jeśli upłynął czas większy niż ustawiony...
break #...następuje wyjście z pętli
fi
# Sprawdzanie czy naciśnięty jest przycisk monostabilny
while [ $(gpio -g read $HALT) -eq 0 ]; do
if [ $(($(date +%s)-currenttime)) -ge 2 ]; then #jeśli jest przytrzymany przez przynajmniej 2 sekundy...
gpio -g write $LED 1
shutdown -h now #...następuje wyłączenie komputera
fi
done
done
#po zakończeniu pętli oczekiwania na odpowiedni odstęp czasowy:
OUTFILE=`printf „$DEST/$PREFIX%05d.jpg” $FRAME` #przygotuj nazwę dla nowego pliku
# echo $OUTFILE
gpio -g write $LED 1 #włącz diodę LED
raspistill -n -w $WIDTH -h $HEIGHT -q $QUALITY -th none -t 250 -o $OUTFILE #wykonaj zdjęcie i zapisz do pliku
gpio -g write $LED 0 # wyłącz diodę LED
FRAME=$((FRAME + 1)) # Zwiększ numer aktualnego zdjęcia
prevtime=$currenttime # Zapisz aktualny czas
done
```