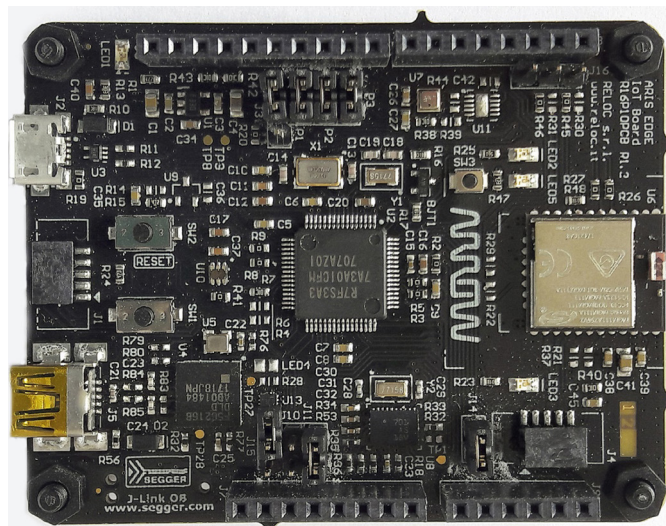


Zestaw startowy Aris Edge S3A3 z mikrokontrolerem Renesas Synergy (1)

Internet Rzeczy czyli IoT to w świecie elektroniki bardzo gorący temat. Producenci komponentów elektronicznych, przede wszystkim mikrokontrolerów, ale również wszelkiego rodzaju czujników, detektorów i modułów komunikacyjnych starają się jak najlepiej odpowiedzieć na dynamicznie rosnące potrzeby projektantów i producentów urządzeń IoT.

Przykładem kompleksowego rozwiązania, które można wykorzystać do projektowania i testowania aplikacji IoT jest niewielki, kompaktowy moduł Aris Edge S3 oferowany przez firmę Arrow. Aris (Arrow Renesas IoT Synergy) jest elastyczną platformą przeznaczoną do projektowania i testowania urządzeń IoT. Głównym elementem modułu jest mikrokontroler Renesas Synergy z rodziny S3 o oznaczeniu R7F-S3A37A3A01CFM. Renesas Synergy S3 jest oparty o rdzeń ARM Cortex M4 taktowany maksymalną częstotliwością 48 MHz i łączy dużą wydajnością obliczeniową ze stosunkowo niskim poziomem poboru energii. Operacje zmiennoprzecinkowe i DSP są wykonywane przy minimalnej ilości cykli zegarowych CPU. Tworzenie interfejsu użytkownika jest wspomagane przez wbudowany sterownik segmentowej matrycy LCD SLCD i sterownik pojemnościowej klawiatury dotykowej. Bezpieczeństwo przechowywania i przesyłania danych zapewniają moduły wykrywania błędów pamięci, układ obliczania CRC, protekcji odczytu pamięci Flash, oraz modułu zabezpieczeń i kodowania nap AES.

Na **rysunku 1** pokazano zasoby procesorów z rodziny S3: pamięci programu i pamięci stałych danych Flash, pamięci danych SRAM i układów peryferyjnych. Zależnie od typu mikrokontrolera mamy do dyspozycji do 1 MB pamięci programu Flash, do 16 kB



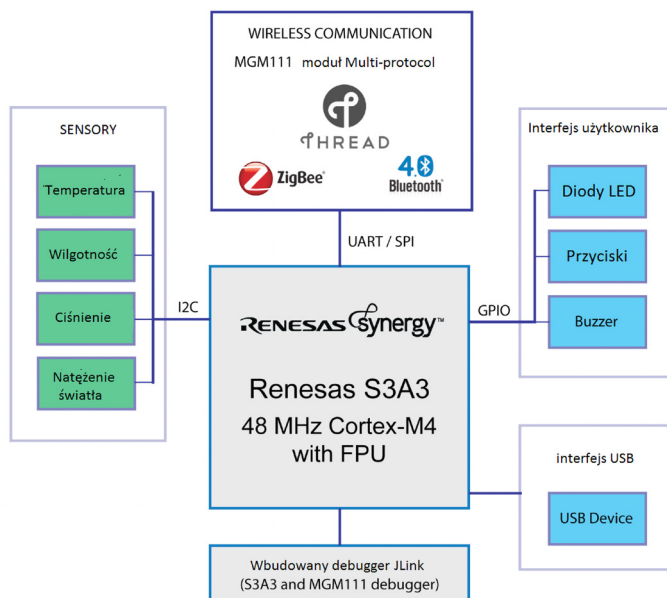
programowanej stałej pamięci Flash przeznaczonej dla danych użytkownika oraz do 192 kB pamięci danych SRAM.

Wydajny i energetycznie oszczędny rdzeń jest obudowany wieloma zaawansowanymi modułami peryferyjnymi. Najważniejsze z nich to:

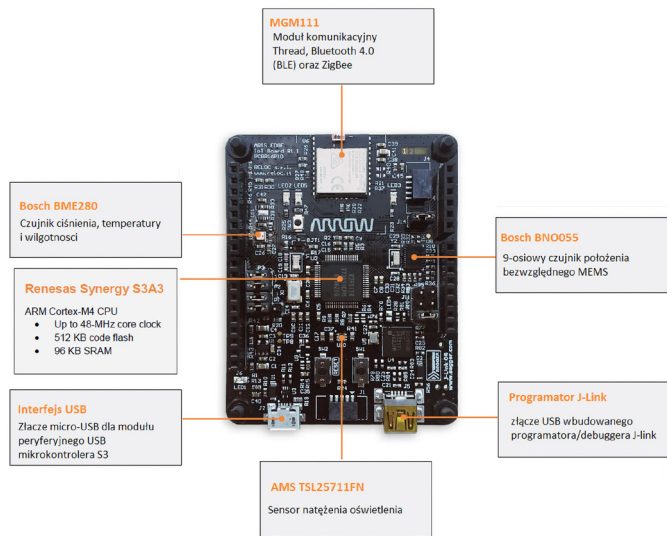
- Moduły analogowe: 14-bitowy przetwornik analogowo-cyfrowy, dwa 12-bitowe przetworniki cyfrowo-analogowe, dwa szybkie i dwa wolne komparatory, cztery wzmacniacze operacyjne i czujnik temperatury.
- Interfejsy komunikacyjne: USB 2.0 Full Speed z implementacją Compilant with USB Battery Charging 1.2, USB2.0 Full Speed z funkcją hosta, 6 szeregowych uniwersalnych kanałów SCI, które można zaprogramować jako USART, I²C (master), SPI,

48 MHz ARM® Cortex®-M4 CPU S3			
FPU ARM MPU NVIC ETB JTAG SWD Boundary Scan			
Memory	Analog	Timing & Control	HMI
Code Flash (up to 1 MB)	14-Bit A/D Converter	General PWM Timer 32-Bit	Capacitive Touch Sensing Unit
Data Flash (up to 16 KB)	12-Bit D/A Converter	Asynchronous General Purpose Timer	Segment LCD Controller
SRAM (up to 192 KB)	Low-Power Analog Comparator	WDT	
Flash Cache	High-Speed Analog Comparator		
Memory Protection Unit	OPAMP		
Memory Mirror Function	Temperature Sensor		
Connectivity	System & Power Management	Safety	Security & Encryption
USBSFS	DMA Controller	ECC in SRAM	128-Bit Unique ID
CAN SDHI/MMC	Data Transfer Controller	SRAM Parity Error Check	TRNG
Serial Communications Interface	Event Link Controller	Flash Area Protection	AES (128/256)
IrDA Interface	Low Power Modes	ADC Diagnostics	GHASH
QSPI SPI	Multiple Clocks	Clock Frequency Accuracy Measurement Circuit	
IIC SSI	Port Function Select	CRC Calculator	
External Memory Bus	RTC	Data Operation Circuit	
	SysTick	Port Output Enable for GPT	
	Low Voltage Detection	IWDT	

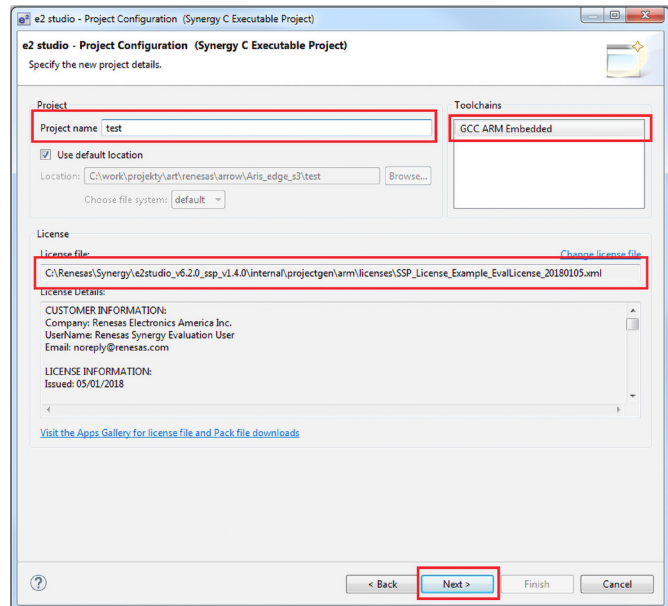
Rysunek 1. Zasoby mikrokontrolerów serii S3



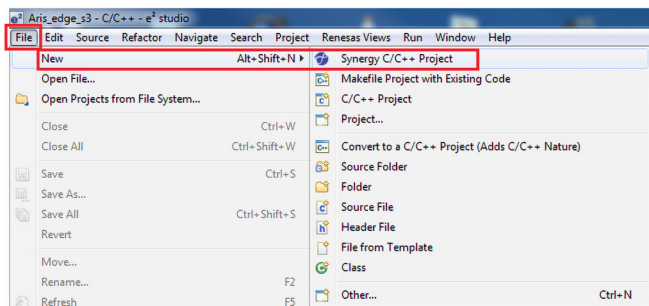
Rysunek 2. Aris Edge S3A3 – struktura funkcjonalna



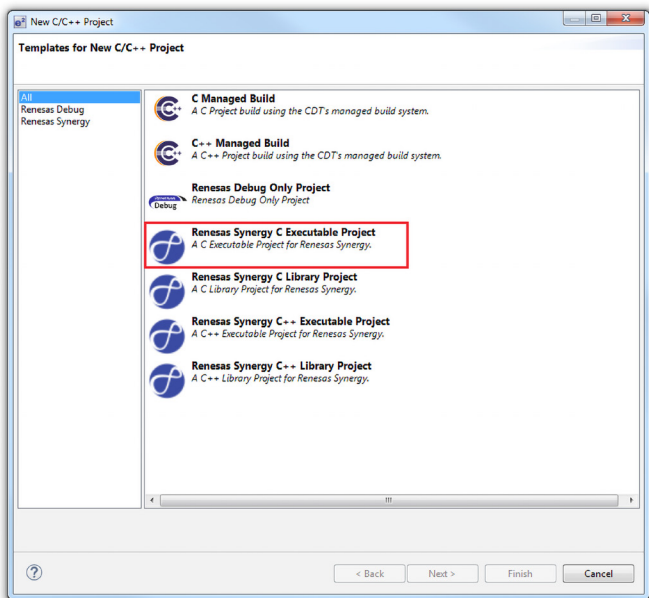
Rysunek 3. Mikrokontroler i zewnętrzne układy peryferyjne na płytce Aris Edge S3A3



Rysunek 6. Nazwa projektu, ścieżki dostępu do projektu i plików licencji



Rysunek 4. Nowy projekt Synergy w środowisku e2studio



Rysunek 5. Wybór rodzaju projektu (Synergy C executable)

lub interfejs Smart Card, 2×SPI, 3×I²C, CAN, Serial Sound Interface (SSI), IrDa, Quad Serial Peripheral Interface (QSPI), host do obsługi transmisji z kartami SD/MMC.

- Moduły czasowe: dziesięć liczników GPT32 z funkcją PWM, dwa liczniki asynchroniczne AGT, licznik watchdog WDT, zegar RTC (z podtrzymaniem baterijnym).
- Moduł HMI: sterowniki wyświetlacza LCD SLDC i układ pojemnościowej klawiatury dotykowej CTSU.
- Układy szyfrowania i bezpieczeństwa: AES128/256, GHASH i generator liczb losowych TRNG.

- Moduł uniwersalnych linii GPIO.
- Mikrokontroler R7FS3A37A3A01CFM zamontowany w module Aris Edge S3A3 jest umieszczony w 64-pinowej obudowie LQFP do montażu powierzchniowego i ma wbudowane 512 KB pamięci programu Flash, 96 kB pamięci danych SRAM i 8 KB stałej pamięci danych Flash. Mamy tu do dyspozycji 52 linie I/O, 6 kanałów uniwersalnego interfejsu komunikacyjnego SCI, po 2 interfejsy SPI i I²C, 14 bitowy przetwornik ADC z maksymalnie 18 kanałami pomiarowymi, 12 bitowy przetwornik DAC, interfejs USBFS, oraz liczniki PWM o rozdzielczości 16 i 32 bity i 4 kanały DMA. Jest też moduł DTC (Data Transfer Controller), który z punktu widzenia użytkownika działa jak kanał DMA obsługujący dużą ilość transferów.

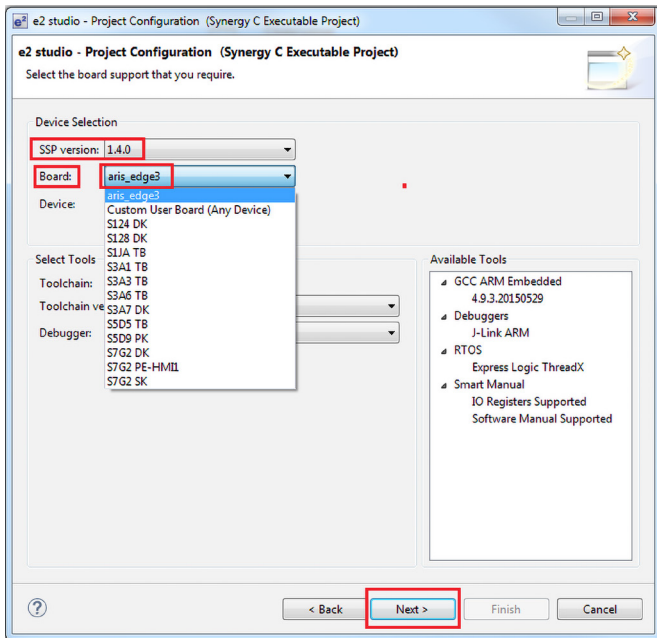
Jak w większości tego typu modułów został wyposażony w układ programatora/debuggera J-Link firmy Segger komunikujący się z komputerem za pomocą interfejsu USB ze złączem mini-USB. J-Link jest wspierany przez środowisko e2 studio bezpłatnie dystrybuowane przez Renesasa.

Pomimo kompaktowych wymiarów 70 mm×53 mm oprócz mikrokontrolera i układu debugera na płytce umieszczono dwa rzędy wyprowadzeń zgodnych ze standardem Arduino, oraz szereg ciekawych układów peryferyjnych:

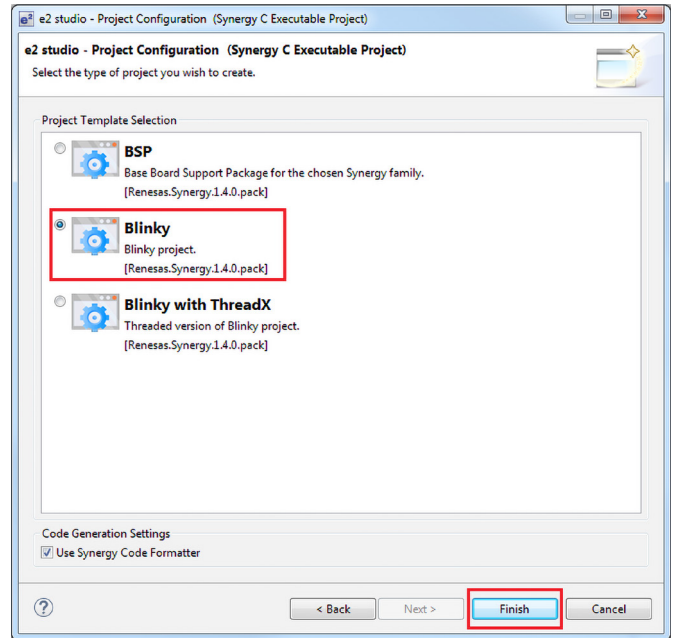
- Moduł komunikacyjny MGM111 realizujący komunikację z wykorzystaniem protokołów Bluetooth Low Energy Thread i ZigBee.
- Czujnik temperatury i wilgotności i ciśnienia atmosferycznego Bosch BME280.
- Czujnik natężenia światła AMS TSL25711FN.
- Czujnik położenia MEMS Bosch BNO055.

Przycisk SW1, dwie diody LED, oraz buzzer mogą być wykorzystane do budowy nieskomplikowanego interfejsu użytkownika. Na **rysunku 2** pokazano schematycznie strukturę budowy modułu z zaznaczeniem rodzajów sensorów i modułów, oraz magistral wykorzystywanych do komunikacji. **Rysunek 3** pokazuje płytkę drukowaną z umieszczeniem układów: mikrokontrolera i układów zabudowanych sensorów.

Jednym z najważniejszych układów peryferyjnych z punktu widzenia aplikacji IoT jest moduł komunikacyjny MGM111umożliwiający implementację bezprzewodowej transmisji w standardzie BLE (Bluetooth 4.0), Thread i ZigBee. Dość istotną właściwością z punktu widzenia uniwersalności jest wspomniane już wyposażenie w złącze standardu Arduino.



Rysunek 7. Wybór wersji biblioteki SSP i płyty Aris Edge 3



Rysunek 8. Wybór szkieletu nowego projektu

```

Listing 1. Procedura sterująca zapaleniem i gaszeniem diod LED modułu
/* File Name : hal_entry.c
 * Description : This is a very simple example application that
 * blinks all LEDs on a board.
 */
#include "hal_data.h"

/* @brief Blinky example application
 * Blinks all leds at a rate of 1 second using the software delay
 * function provided by the BSP. Only references two other modules
 * including the BSP, IOPORT.
 */
void hal_entry(void)
{
    /* Define the units to be used with the software delay function */
    const bsp_delay_units_t bsp_delay_units = BSP_DELAY_UNITS_MILLISECONDS;
    /* Set the blink frequency (must be <= bsp_delay_units */
    const uint32_t freq_in_hz = 2;
    /* Calculate the delay in terms of bsp_delay_units */
    const uint32_t delay = bsp_delay_units/freq_in_hz;
    /* LED type structure */
    bsp_leds_t leds;
    /* LED state variable */
    ioport_level_t level = IOPORT_LEVEL_HIGH;
    /* Get LED information for this board */
    R_BSP_LedsGet(&leds);
    /* If this board has no LEDs then trap here */
    if (0 == leds.led_count)
    {
        while(1); // There are no LEDs on this board
    }
    while(1)
    {
        /* Determine the next state of the LEDs */
        if(IOPORT_LEVEL_LOW == level)
        {
            level = IOPORT_LEVEL_HIGH;
        }
        else
        {
            level = IOPORT_LEVEL_LOW;
        }
        /* Update all board LEDs */
        for(uint32_t i = 0; i < leds.led_count; i++)
        {
            g_ioport.p_api->pinWrite(leds.p_leds[i], level);
        }
        /* Delay */
        R_BSP_SoftwareDelay(delay, bsp_delay_units);
    }
}

```

```

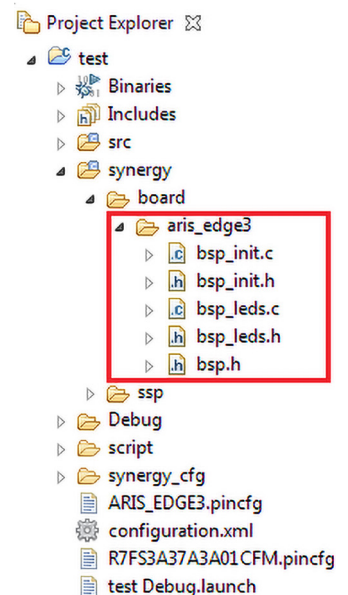
Listing 2. Wsparcie użycia diod LED z pliku bsp_leds.c
/* Private global variables and functions */
/* Array of LED IOPORT pins. */
static const ioport_port_pin_t g_bsp_prv_leds[] =
{
    IOPORT_PORT_03_PIN_03, ///< LED2
    IOPORT_PORT_01_PIN_13, ///< LED3
};
/* Exported global variables (to be accessed by other files) */
/* Structure with LED information for this board. Recommended to get this information through R_BSP_LedsGet() function
 * instead of using this structure directly to remove dependency on structure name. */
const bsp_leds_t g_bsp_leds =
{
    .led_count = (uint16_t)((uint16_t)sizeof(g_bsp_prv_leds) / (uint16_t)sizeof(g_bsp_prv_leds[0])),
    .p_leds = &g_bsp_prv_leds[0]
};

```

Jak widać budowa modułu jest przemyślana i ma wiele zalet:

- Kompaktowa, zwarta obudowa, w której umieszczono złącza standardu Arduino.
- Wbudowany programator/debugger Segger J-link.
- Wydajny, oszczędny mikrokontroler Renesas Synergy z rodziny S3.
- Umieszczenie na płytce zaawansowanych układów peryferyjnych i złącza USB Host.

Sprzętowo płytka jest przygotowana do testowania aplikacji IoT. Jednak sam sprzęt to nie wszystko. Renesas oferuje bardzo silne wsparcie narzędziowe dla programistów. Można używać bezpłatnego świetnego firmowego pakietu IDE znanego jako e2-studio zintegrowanego z firmowym pakietem bibliotek SSP. Od niedawna jest dostępny również bezpłatny pakiet IAR Embedded Workbench przeznaczony dla projektów



Rysunek 9. Pliki związane z wyborem płyty aris_edge3

z mikrokontrolerami Renesas Synergy. Pakiet IAR współpracuje z samodzielnym konfiguratorem SSC dystrybuowanym przez Renesas. To wsparcie powoduje, że IAR Workbench ma takie same możliwości używania bibliotek SSP i ich konfigurowania jak to jest możliwe w e2studio.

W sieci jest dostępnych wiele programów przykładowych i not aplikacyjnych pozwalających na bezproblemowe rozpoczęcie pracy z mikrokontrolerami Renesas Synergy. Dla opisywanego modułu firma Reloc przygotowała wsparcie projektów tworzonych w środowisku e2studio w postaci pliku z rozszerzeniem .PACK. Pliki PACK są integralnie związane z wersją biblioteki SSP i należy ich używać w projektach z wybraną właściwą wersją SSP. Dla modułu w momencie pisania artykułu były dostępne pliki *Reloc.Aris_Edge3.1.3.3.PACK* przeznaczony dla wersji SSP V1.3.3 i *Reloc.Aris_Edge3.1.4.0.PACK* przeznaczony dla wersji SSP V1.4.0. Użycie tych plików pokażemy na podstawie przykładowego projektu środowiska e2studio v 6.2.0 i biblioteki SSP V1.4.0.

W pierwszym kroku trzeba skopiować plik *Reloc.Aris_Edge3.1.4.0.PACK* do katalogu `RenasantSynergy\e2studioV6.2.0\internal\projectgen\arm\packs`. Po uruchomieniu e2 studio tworzymy nowy

projekt z menu narzędziowego *File→New→Synergy C/C++ Project (rysunki 4..6)*. Na **rysunku 7** pokazano kolejne okno konfiguracji projektu, w którym wybiera się wersję zainstalowanej biblioteki SSP – w naszym przypadku V1.4.0 i wsparcie dla płyty Aris Edge 3. Możliwość wyboru *aris_edge3* z rozwijanego okna *board* pojawi się po skopiowaniu pliku *Reloc.Aris_Edge3.1.4.0.PACK* do opisanej powyżej lokalizacji. Ostatnim krokiem generowania projektu jest wybór rodzaju szkieletu. Do celów testowych możemy wybrać opcję *Blinky (rysunek 8)*. Wygenerowany przez kreatora projekt będzie cyklicznie zapalał i gasił diody LED2 i LED3 umieszczone na płycie. Na **listingu 1** pokazano procedurę *hal_entry.c* sterująca cyklicznym zapaleniem i gaszeniem diod LED D2 i D3.

Wybór *aris_edge3* w oknie Board powoduje, że konfigurator projektu umieści kilka plików dodatkowych w folderze *aris_edge3* – zostało to pokazane na **rysunku 9**. Poza definiowaniem diod LED D2 i D3 (**listingu 2**) nie znajdziemy tam zbyt wielkiego wsparcia dla układów peryferyjnych. Dlatego należy je traktować jako szkielet wypełniany przez użytkownika zależnie od potrzeb.

Tomasz Jabłoński, EP

REKLAMA

m.technik
Ciekawi świata są zawsze młodzi

w prezencie na każdą okazję

BRACIA KASHIO
Cztery jeźdźcy japońskiej rewolucji
nr 9. wrzesień 2018
e-suplement www.mt.com.pl

m.technik
Ciekawi świata są zawsze młodzi

co z tą
ELEKTRONIKĄ?
czy dotarła do kresu rozwoju

RAPORT: Policja hi-tech
Science fiction wkracza do komisariatu

przejrzyj i kupisz na
www.ulubionykiosk.pl

<https://goo.gl/TiDLmR>