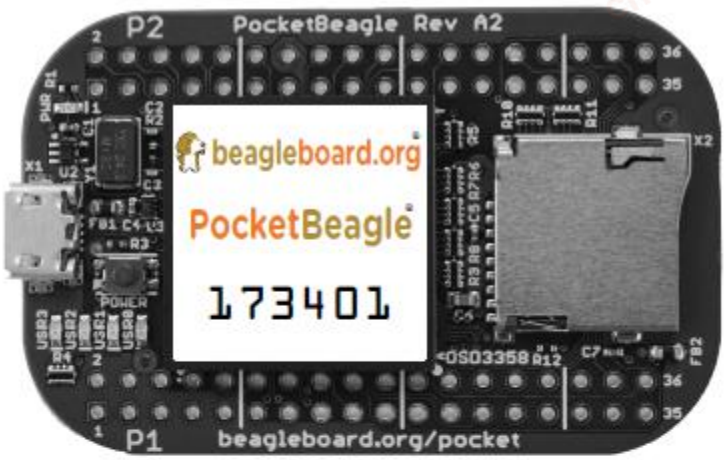
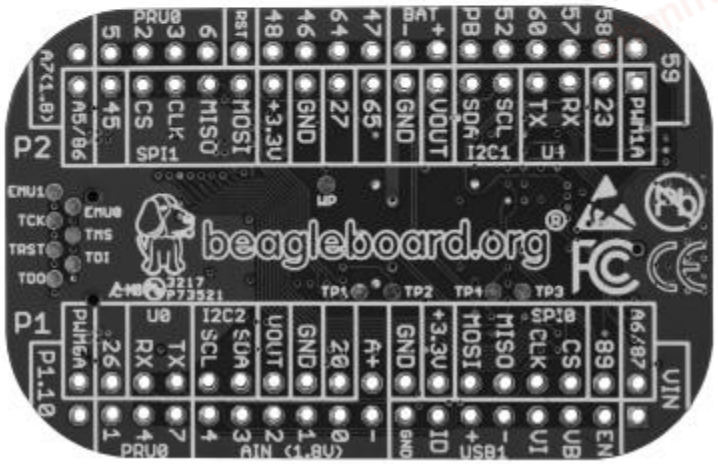
PocketBeagle®

PRODUCT ID: POCKETBEAGLE





DESCRIPTION

The newest BeagleBoard.org® board is PocketBeagle®, an ultra-tiny-yet-complete Linux-enabled, community-supported, open-source USB-key-fob computer. PocketBeagle® features an incredible low cost, slick design and simple usage, making PocketBeagle® the ideal development board for beginners and professionals alike. You develop directly in a web browser and PocketBeagle® can easily be set back to factory conditions, leaving you free to experiment.

KEY FEATURES

- Low cost Linux computer with tremendous expansibility
- Opportunity to learn many programming aspects from educators on-line
- Openness and flexibility tear-down limits on your imagination

What is a USB key-fob computer?

PocketBeagle® is the size of a tiny mint-tin (35mm by 55mm), less than half the size of the larger mint-tin or credit-card sized BeagleBone® Black (55mm by 86mm). Unlike a desktop computer where you connect a monitor, keyboard and mouse, PocketBeagle® is made to live inside your project and enables you to define its interfaces. PocketBeagle® is easily programmed through a web browser running on any other connected desktop.

What can I do with PocketBeagle®?

Getting to the Linux command-line and text editor via your web browser is simple, providing you with a playground for programming and electronics. Exploring is made easy with several available libraries and tutorials and many more coming. Once you get a bit familiar with Linux and electronics, you are free to explore numerous more advanced projects from the community.

The sky is no limit; PocketBeagle® makes a great starting point for building something as advanced as a computer for a CubeSat and there are several BeagleBone® Black examples out there already today. Flying a bit lower, PocketBeagle® is a good target for flight controller software, such as ArduPilot, similar to what is done on BeagleBone® Blue but with the flexibility of choosing all your own sensors and interconnects. Touching the ground, the combination of a 1-GHz Linux computer and 2 powerful 200-MHz hard-real-time shared-memory programmable real-time (PRU) microcontrollers makes driving robotic machines like 3D printers, CNC mills and laser cutters fast and simple with software such as Redeem, MachineKit or BeagleG as great starting points. If you'd like your ground-based machine to talk back to the cloud, the SPI, USB and UART expansion makes adding your own Ethernet, WiFi, Bluetooth and long-range wireless connectivity easy with Linux drivers and Node.JS or Python libraries to add smarts. If what you want is just fun, add a SPI-based display and run off of a single-cell LiPo battery to create your own custom gaming device with the sensors, such as cameras and software like OpenCV or just simple accelerometers, of your own choice to go on an adventure of your own.



What are Programmable Real-Time Units (PRUs) anyway?

Texas Instruments' Sitara AM335x processor is built of several central processing units (CPUs), including two programmable real-time units (PRUs). These are 200MHz 32-bit microcontroller CPUs with a zero-depth pipeline and single-cycle access to a collection of pins and peripherals optimized for implementing software-defined peripherals. The PRUs don't have pre-defined tasks in the system, so they are free to run your software. They are ideal for tight control loops; a must-have feature for building quadcopters, 3D printers and balancing robots, just to name a few.

PRUs are ideal for predictable low-latency, whereas the ARM processor is good for throughput. Latency is how quickly you can respond. Throughput is how quickly you can move once you reach top speed. When you need to handle a lot of little tasks, you need low latency and low overhead. When you have a large payload to process, you want good throughput. PocketBeagle® is designed to be good at both and efficient in enabling them to work together.

Why is PocketBeagle® good for a novice?

PocketBeagle® is affordable, enabling you to dedicate one to live permanently in each of your different projects. Even though PocketBeagle® is very low-cost, it is made with top quality engineering and manufacturing. In the unlikely event it is damaged due to misuse, it won't cost much to replace.

PocketBeagle® will boot directly from on-board ROM that cannot be accidentally modified and will load software via USB, serial or microSD cards. A Chrome plug-in or cross-platform Node.JS Electron app can boot your board to add a Linux distribution to an attached microSD card. This means you can create reliably reproducible instructions on using the board, because it will behave the same way every time.

PocketBeagle® runs GNU/Linux, which means you can leverage many different high-level programming languages and a large body of drivers that prevent you from needing to write a lot of your own software.

PocketBeagle® doesn't require you to install a bunch of tools on your host computer. You develop directly in your web browser, a tool you already know, and most users won't ever even need administrator privileges on their host computer.

Why is PocketBeagle® good for a professional?

PocketBeagle® utilizes a simple open-source hardware design making it easy for you to take control of your own destiny, either customizing the board or sourcing and manufacturing it in any method you'd like. The processor is well-documented and available broadly, avoiding lurking issues that might otherwise be difficult to resolve. PocketBeagle® avoids consuming precious hardware resources, leaving them available for your design goals while still exposing features for expansion.



Most importantly, PocketBeagle® respects your time and helps you avoid risk by providing a solution for rapid prototyping without slowing your path to production. BeagleBoard.org's community of thousands of serious developers means that someone has probably already tried to do something similar and will be willing to share answers and experiences. PocketBeagle® avoids re-inventing the wheel, just makes it smaller, simpler and more affordable while keeping it flexible. PocketBeagle® support is pushed upstream in Linux and u-boot such that you'll always be able to leverage the latest features and bug fixes in those projects.

Why does PocketBeagle® really matter anyway?

The BeagleBoard.org Foundation has the goal of increasing computer, electronics and robotics literacy at a point they are critical in human history. PocketBeagle® increases access to inspirational hardware and we believe hands-on experience is the best way to build understanding and empowerment. We also believe in empowerment at all ages and experience, so we give you the tools to take our designs forward.

How did we make PocketBeagle® so small and affordable?

PocketBeagle® is built around Octavo Systems' System-In-Package that integrates a high-performance TI AM3358 processor, 512MB of DDR3, power management, nonvolatile serial memory and over 140 passive components into a single package. This integration saves cost and a small amount of power by eliminating several packages that would otherwise need to be placed on the board, but more notably it saves the space of all those packages and simplifies our board design so we can focus on the user experience.

SPECIFICATIONS

Texas Instruments® Sitara™ AM3358 Processor (Integrated in the OSD3358-SM):

- 1GHz ARM® Cortex-A8 with NEON floating-point accelerator
- SGX530 graphics accelerator
- 2x programmable real-time unit (PRU) 32-bit 200MHz microcontrollers with single-cycle I/O latency
- ARM® Cortex-M3 for power and security management functions

Memory:

- 512MB DDR3 800MHZ RAM (Integrated in the OSD3358-SM)
- 4kB I2C EEPROM (Integrated in the OSD3358-SM)
- SD/MMC Connector for microSD

Software Compatibility

- Debian GNU/Linux images customized for BeagleBone
- Cloud9 IDE on Node.js w/ BoneScript library
- Any BeagleBone Black software not needing access to unavailable expansion pins



Connectivity

- High speed USB 2.0 OTG (host/client) micro-B connector (USB0)
- Bootable microSD card slot (MMC0)

Expansion header

- High speed USB 2.0 OTG (host/client) control signals (USB1)
- 8 analog inputs with 6 at 1.8V and 2 at 3.3V along with 1.8V voltage references
- 44 digital GPIOs accessible with 18 enabled by default including 2 shared with the 3.3V analog input pins
- 3 UARTs accessible with 2 enabled by default (UART0, UART4)
- 2 I2C busses enabled by default (I2C1, I2C2)
- 2 SPI busses with single chip selects enabled by default (SPI0, SPI1)
- 4 PWM outputs accessible with 2 enabled by default (PWM0A, PWM1A)
- 2 quadrature encoder inputs accessible
- 2 CAN bus controllers accessible
- 23 programmable real-time unit (PRU) 32-bit microcontroller I/O pins including options for the PRU UART and eCAP accessible with 7 I/O pins enabled by default for PRU0 and 1 enabled by default for PRU1
- 3 voltage inputs with 1 for battery, 1 shared with the USB connector and 1 for power-line input and a battery temperature sense input
- 2 voltage outputs, 1 with a 3.3V LDO and 1 with switch from voltage input
- Power and reset button I/Os

Power management:

- TPS65217C PMIC is used along with a separate LDO to provide power to the system (Integrated in the OSD3358) with integrated 1-cell LiPo battery support

Debug Support:

- JTAG test points
- gdb and other monitor-mode debug possible

Power Source

- microUSB connector
- expansion header (3 options: battery, VIN or USB-VIN)

User Input / Output

- Power Button with press detection interrupt via TPS65217C PMIC (hold for 10s to initiate hardware power cycle)

beagleboard.org®

