Porting of MicroPython to LEON Platforms

 $\nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B} / \partial t$

 $J + 1/c^2 \partial E / \partial t F = q (E + \mathbf{v} \times B) - \hbar^2 / 2m \nabla^2 \psi$

Damien P. George

George Robotics Limited, Cambridge, UK

TEC-ED & TEC-SW Final Presentation Days ESTEC, 10th June 2016

$$\begin{split} & d + 1/2 \frac{\partial d U}{\partial t} (\beta t + a + (\mathbb{R} + \mathbf{x} - \mathbb{R}) - h^2 2 m^2 2 a(\mathbf{x}, 1) + V(\mathbf{x}) a(\mathbf{x}, 1) = E a(\mathbf{x}, 1) d^2 - a(\mathbf{x} - 1) t^2 - a(\mathbf{x} - 1$$

 $\begin{aligned} & \int_{\mathbb{R}^{2}} (p_{1}p_{1}p_{2}) p_{2}p_{1}p_{2}p_{2} = \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2}p_{2} = p_{2}p_{1}^{2} \\ & \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2}p_{2} = p_{2}p_{2}^{2} \\ & \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2}p_{2} = p_{2}p_{2}^{2} \\ & \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2}p_{2} = p_{2}p_{2}p_{2}^{2} \\ & \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2}p_{2} \\ & \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2} \\ & \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2}p_{2} \\ & \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2} \\ & \int_{\mathbb{R}^{2}} (p_{2}p_{2}) p_{2$

George Robotics Limited (UK)

- a limited company in the UK, founded in January 2014
- specialising in embedded systems
- hardware: design, manufacturing (via 3rd party), sales
- $v E = \gamma mc^2 E^2 = p^2 c^2 + m^2 c^4 \partial_\mu J^\mu = 0 E_i = -1/c \partial A_i$ software: development and support of MicroPython code $\delta = 2GM/R \int \psi^* \psi dx = 1 P(x, t) = |\psi(x, t)|^2 \psi(x, t) = |\psi(x, t)|^2$

 $\nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E} / \partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi$

2/21

 $d^{2} = (1 - 2GM/r)dt^{2} - dr^{2}/(1 - 2GM/r) - r^{2}d\theta^{2} - r^{2}\sin^{2}\theta d\phi^{2} \Delta \nu \approx 10^{10}$

 $U = e^{Ht/1\hbar} \mathbf{F} = m\mathbf{a} \mathbf{F} = GMm\mathbf{r}/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

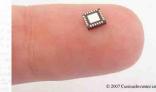
 $/ds + \Gamma^{\mu}_{\nu\sigma} v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\nu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma} \Gamma^{\mu}_{\rho,\rho} - \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\sigma} \ R_{\mu\nu} = R^{\mu}_{\mu\nu\rho} \ R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$ $R \int \psi^a \psi dx = 1 P(x,t) = |\psi(x,t)|^2 |\psi(x,t) = |\psi(t)\rangle \langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \ge \hbar/2 | p_{\perp} = -i\hbar\partial_{\perp} E = i\hbar\partial/\partial t | H = p^2/2m + V$ $x)\psi(x, t) = E\psi(x, t) x' = \gamma(x - vt) t' = \gamma(t - vx/c^2) \gamma = 1/(1 - v^2/c^2)^{1/2} x' = -t \sinh \xi + x \cosh \xi t' = t \cosh \xi - x \sinh \xi t t'$ $\lambda_{k} F^{\mu\nu} = \partial^{\mu} \lambda^{\nu} - \partial^{\nu} \lambda^{\mu} \bar{F}^{\mu\nu} = 1/2 \epsilon^{\mu\nu\rho\sigma} F_{\rho\sigma} \partial_{\nu} F^{\mu\nu} = J^{\nu} \partial_{\nu} \bar{F}^{\mu\nu} = 0 \, ds^{2} = c^{2} dt^{2} - dx^{2} - dx^{2} - dx^{2} \, ds^{2} = g_{\mu\nu} dx^{\mu} dx^{\nu} dx^{\nu}$ $R = R_{\mu}^{\mu} G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = 0 \ G_{\mu\nu} = 8\pi G T_{\mu\nu} \ ds^2 = (1 - 2GM/r)dt^2 - dr^2/(1 - 2GM/r) - r^2 d\theta^2 - r^2 \sin^2\theta d\phi^2 \ \Delta\nu \approx 10^{-10} \ ds^2 + 10^{-10} \ ds^2 = 10^{-10} \ ds^2 + 10^{-10} \ ds^2 + 10^{-10} \ ds^2 = 10^{-10} \ ds^2 + 10^{-10} \ ds^2 = 10^{-10} \ ds^2 + 10$ $\hbar/2|p_1 = -\hbar\hbar\partial_1|E| = \hbar\hbar\partial/\partial t|H| = p^2/2m + V|H|a| = E|a||U| = e^{Ht/\hbar\hbar}|\mathbf{F}| = m\mathbf{a}|\mathbf{F}| = GMm\mathbf{r}/r^3|\nabla\cdot\mathbf{E}| = \rho/\epsilon|\nabla\cdot\mathbf{B}| = 0|\nabla\times\mathbf{E}| = -\partial\mathbf{B}/\partial t||\mathbf{F}| = h/\epsilon |\mathbf{F}||\mathbf{F}| = h/\epsilon |\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F}||\mathbf{F$ $= J^{\nu} \, \beta_{\nu} F^{\mu\nu} = 0 \, \mathrm{d} s^{2} = c^{2} \mathrm{d} t^{2} - \mathrm{d} x^{2} - \mathrm{d} y^{2} - \mathrm{d} z^{2} \, \mathrm{d} s^{2} = g_{\mu\nu} \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \, g_{\mu\nu} (\mathrm{d} x^{\mu}/\mathrm{d} r) \\ = 1 \, \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \, \mathrm{d} x^{\mu} \, \mathrm{d} x^{\nu} \, \mathrm{d} x^{\nu$

 $E[a] U = e^{Ht/\hbar} F = \operatorname{ina} F = GMmr/r^3 \nabla \cdot E = \rho/\epsilon \nabla \cdot B = 0 \nabla \times E = -\partial B/\partial t \nabla \times B = \mu J + 1/c^2 \partial E/\partial t F = g(E + \mathbf{v} \times B) - \hbar^2/2m\nabla^2 \psi$ $\tanh \xi = v/c \, \cosh \xi = \gamma \, L = L_0 / \gamma \, T = \gamma T_0 \, u' = (u - v) / (1 - uv/c^2) \, p = \gamma mv \, E = \gamma mc^2 \, E^2 = p^2 c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^4 \, \partial_{\mu} J^{\mu} = 0 \, E_4 = -1/c \partial A_4 / c^2 + m^2 c^$ $\mathrm{d} x^2 - \mathrm{d} x^2 - g_{\mu\nu} \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} - g_{\mu\nu} \mathrm{d} x^{\mu} \mathrm{d} r) (\mathrm{d} x^{\mu} / \mathrm{d} r) (\mathrm{d} x^{\nu} / \mathrm{d} r) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \ \mathrm{d} v^{\mu} / \mathrm{d} s + \Gamma^{\mu}_{\mu\sigma} v^{\nu} v^{\sigma} = 0 \ R^{\mu}_{\nu\rho\sigma} - \Gamma^{\mu}_{\nu\rho,\sigma} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma} + \Gamma^{$ $=r^{2}d\theta^{2} - r^{2}\sin^{2}\theta d\phi^{2} \quad \Delta r \approx \nu_{c}GM(1/r_{c} - 1/r_{c}) \quad d^{2}u/d\phi^{2} + u - GM/A^{2} - 3GMu^{2} = 0 \quad \delta = 2GM/R \quad \left[\psi^{*}\psi dx = 1 \quad P(x, t) = |\psi(x, t)|^{2} \quad \psi(x, t) = |\psi(x, t)|^{2} \quad$ $\nabla \cdot \mathbf{B} = 0 \ \nabla \times \mathbf{E} = -\partial \mathbf{B} / \partial t \ \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E} / \partial t \ \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x}) \psi(\mathbf{x}, t) = E \psi(\mathbf{x}, t) \ x' = \gamma(x - vt) \ t' = \gamma(t - v) + \frac{1}{2} (x - v)$ $=\psi)/(1-u\nu/c^2) \ p = \gamma m\nu \ E = \gamma mc^2 \ E^2 = p^2c^2 + m^2c^4 \ \partial_\mu J^\mu = 0 \ E_i = -1/c\partial A_i/\partial t - \partial_i\phi \ B_i = \epsilon_{ijk}\partial_iA_k \ F^{\mu\nu} = \partial^\mu A^\mu \ \bar{\rho}^{\mu\nu} = 1/2\epsilon^{\mu} + 1/$ $1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) dv^{\mu}/ds + \Gamma^{\mu}_{\nu\sigma}v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\nu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma}\Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\nu\rho}\Gamma^{\mu}_{\alpha\sigma} R_{\mu\nu} = R^{\mu}_{\mu\nu\rho} R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$ $\mathrm{d}\phi^2 + u = GM/A^2 = 3GMu^2 = 0 \ \delta = 2GM/R \ \int \psi^* \psi \mathrm{d}x = 1 \ P(x,t) = |\psi(x,t)|^2 \ \psi(x,t) = |\psi(t)\rangle \ \langle x \rangle = \langle \psi | x | \psi \rangle \ \Delta x \Delta p \ge \hbar/2 \ p_i = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = p_i \ A = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ H = -i\hbar\partial/\partial t \ H = -$ D.P. George (George Robotics Ltd) MicroPython on LEON

Motivation for MicroPython

Electronics circuits now pack an enormous amount of functionality in a tiny package.

Need a way to control all these sophisticated devices.



+ $V |I|_{\alpha}$) - $E|_{\alpha}$) $U = e^{Ht/i\hbar}$ F = ma F = $GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

Scripting languages enable rapid development. $g_{\mu\nu}(dx^{\mu}/d\tau)(dx^{\nu}/d\tau) = 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$ $\sin^2 \theta d\phi^2 \Delta \nu \approx \nu_i GM(1/r_i - 1/r_f) d^2 u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \delta =$ $\mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/\epsilon^2 \partial \mathbf{E}/\partial t \mathbf{F} = g(\mathbf{E} + \mathbf{y} \times \mathbf{B}) - \hbar^2/2m\nabla^2 g$ Is it possible to put Python on a microcontroller?

 $\mathbf{E} = p/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar$

Why is it hard?

 $\xi = x \sinh \xi \tanh \xi = v/c \cos \theta$ Very little memory (RAM, ROM) $GM/r_1 = r^2 d\theta^2 = r^2 \sin^2 \theta d\phi^2 \Delta$ on a microcontroller.



 $/\delta\phi^2 + u = GM/A^2 - 3GMu^2 = 0 \delta = 2GM/R \int \psi^* \psi dx = 1 P(x,t) = |\psi(x,t)|^2 \psi(x,t) = |\psi(t)\rangle \langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \geq h/2 \ p_i = -ih\partial_i \ E = ih\partial_i \partial t \ H = p_i \langle x, t \rangle = h/2 \ h/2$ D.P. George (George Robotics Ltd)

MicroPython on LEON

 $\partial B/\partial t \nabla \times B = \mu J + 1/c^2 \partial E/\partial t F = a (E + y \times B) - \hbar^2/2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x})\psi(\mathbf{x}, t) = E\psi(\mathbf{x}, t) \mathbf{x}' = \gamma(\mathbf{x} - \mathbf{y}t) \mathbf{t}' = \gamma(t - \mathbf{y})$

Why Python?

- High-level language with powerful features (classes, list comprehension, generators, exceptions, ...).
- Large existing community.
- Very easy to learn, powerful for advanced users: shallow but long learning curve.
- Ideal for microcontrollers: native bitwise operations, procedural code, distinction between int and float, robust exceptions.
- Lots of opportunities for optimisation (this may sound surprising, but Python is compiled).

$$\begin{split} & (1 + 1) \sum_{\mu = 0}^{\mu = 0} e^{-\frac{1}{2}} e^{-\frac{1}{2}}$$

D.P. George (George Robotics Ltd)

MicroPython on LEON

4/21

 $= GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

 $= \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2/2m\nabla^2 \psi$

Why can't we use existing CPython? (or PyPy?)

Integer operations:

Integer object (max 30 bits): 4 words (16 bytes) Preallocates 257+5=262 ints \longrightarrow 4k RAM! Could ROM them, but that's still 4k ROM. And each integer outside the preallocated ones would be another 16 bytes.

 $B = \mu J + 1/c^2 \partial E / \partial t F = q (E + v \times B) - \hbar^2 / 2m \nabla^2 \psi$

 $\begin{array}{l} 3 + i/c^2 \partial E/\partial \mathbf{F} = q \left(E + \mathbf{v} \times \mathbf{B} \right) - \hbar^2/2m\nabla^2 \psi(\mathbf{x},t) + V(\mathbf{x})\psi(\mathbf{x},t) = E\psi(\mathbf{x},t) \, \mathbf{x}' = \gamma(\mathbf{x} - \mathbf{x}) \, \mathbf{x}' = \gamma(\mathbf{x} - \mathbf{x}) \, \mathbf{x}' \\ 2^2 c^2 + m^2 c^4 \, \partial_\mu J^\mu = 0 \, E_i = -1/c \partial A/(\partial t - \partial_\mu B_i = \epsilon_{ijk} \partial_j A_k \, F^{\mu\nu} = \partial^\mu A^{\mu\nu} - \partial^{\mu} A^{\mu\nu} - h^{\mu\nu} I^{\mu\nu} = 1/2 \, \mathbf{x} \\ = 0 \, h^2_{cons} = t^{\mu}_{cons} - t^{\mu}_{cons} - t^{\mu}_{cons} f^{\mu\nu}_{cons} - t^{\mu\nu}_{cons} f^{\mu\nu}_{cons} = R^{\mu}_{\mu\nu} B_i = R^{\mu}_{\mu} \, \mathbf{G}_{\mu\nu} = R_{\mu\nu} - 1/2 \, \mathbf{g}_{\mu\nu} R = t \\ \end{array}$

Method calls:

led.on(): creates a bound-method object, 5 words (20 bytes) led.intensity(1000) \longrightarrow 36 bytes RAM!

For loops: require heap to allocate a range iterator.

 $\begin{aligned} & (z^{-1} - dz^{-1})^{-1} e^{-z_{0}} dz^{-1} e^{-z_{0}} dz^{-1} (dz^{-1})(dz^{-1}/dz^{-1}) = 1 - 1/2(p_{0}p_{0}, z - p_{0}, y_{0} - p_{0}, z_{0} - p_{0}) dz^{-1} dz^{-1} dz^{-1} dz^{-1} dz^{-1} dz^{-1} dz^{-1} (dz^{-1}/dz^{-1}) = 0 \\ & (z^{-1} - z^{-1})^{-1} dz^{-1} dz^{-1}$

Crowdfunding via Kickstarter

Kickstarter, since 2009; collected so far over US\$1 billion in funds

- ▶ 30th April 2013: start!
- 17th September: flashing LED with switch in bytecode Python.
- 21st October: REPL, filesystem, USB VCP and MSD on PYBv2.



 $ma \mathbf{F} = GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$ weekend to make the video. $dx^{\mu}/d\tau$ $(dx^{\nu}/d\tau) = 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$ $\mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = g (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2/2m\nabla^2 \psi$ Kickstarter launched on 13 November 2013, ran for 30 davs. $A_{\mu\nu}^{\mu} R_{\mu\nu} = R_{\mu\nu\rho}^{\rho} R = R_{\mu}^{\mu} G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$ $\Delta \pi \Delta p \ge \hbar/2 p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t H = p^2/2m + V$ $-vx/c^2$ $\gamma = 1/(1-v^2/c^2)^{1/2} x' = -t \sinh \xi + x \cosh \xi t' = t \cosh \xi - x \sinh \xi$ to $\partial_{\nu} \bar{F}^{\mu\nu} = 0 ds^2 = c^2 dt^2 - dx^2 - dy^2 - dz^2 ds^2 = g_{\mu\nu} dx^{\mu} dx^{\nu}$ Total backers: 1,931 Total raised: £97,803 Officially finished 12 April $\sigma_{\sigma,\mu}) dv^{\mu}/ds + \Gamma^{\mu}_{\mu\sigma}v^{\nu}v^{\sigma} = 0 R^{\mu}_{\mu\sigma\sigma} = \Gamma^{\mu}_{\mu\sigma,\sigma} - \Gamma^{\mu}_{\mu\sigma,\sigma} + \Gamma^{\alpha}_{\alpha\sigma}\Gamma^{\mu}_{\alpha\sigma} - \Gamma^{\alpha}_{\mu\sigma}\Gamma^{\mu}_{\alpha\sigma} R_{\mu\nu} = R^{\rho}_{\mu\nu\sigma} R = R^{\mu}_{\mu} G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$

 $+1/c^2 \partial E/\partial t F = a(E + y \times B) - \hbar^2/2m\nabla^2 \psi$

D.P. George (George Robotics Ltd)

e Robotics Ltd) MicroPython on LEON 6/21

Manufacturing

Jaltek Systems, Luton UK — manufactured 13,000+ boards.



 $s^2 = g_{\mu\nu} dx^{\mu} dx^{\nu} g$ 0B/01 $+g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$ (B) $-\hbar^2/2m\nabla^2\psi$

 $\mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B} / \partial t$



 $-r^{2}\sin^{2}\theta d\phi^{2} \Delta v \approx v_{c}GM(1/r_{c}-1/r_{c}) d^{2}u/d\phi^{2} + u - GM/A^{2} - 3GMu^{2} = 0 \delta = 2GM/R \left[\psi^{*}\psi dx = 1 P(x,t) = |\psi(x,t)|^{2} \psi(x,t) =$ $\mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = q (\mathbf{E} + \mathbf{y} \times \mathbf{B}) - h^2/2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x})\psi(\mathbf{x}, t) = E\psi(\mathbf{x}, t) \mathbf{x}' = \gamma(x - vt) \mathbf{t}' = \gamma(t - vt) \mathbf{x}' + \frac{1}{2} \nabla \mathbf{x}' + \frac{$



 $uv/c^2) p = \gamma mv E = \gamma mc^2 E^2 = p^2 c^2 + m^2 c^4 \partial_{\mu} J^{\mu} = 0 E_i = -1/c \partial A_i / \partial t - \partial_i \phi B_i = \epsilon_{ijk} \partial_i A_k F^{\mu\nu} = \partial^{\mu} A^{\nu} - \partial^{\nu} A^{\mu} \tilde{F}^{\mu\nu} = 1/2 \epsilon^{\mu} \partial_i A_i F^{\mu\nu} = 0$ $+ g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) d\nu^{\mu}/dx + \Gamma^{\mu}_{\mu\sigma} v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\nu\rho\sigma} - \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\sigma} \ R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} \ R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu} R = 0 \ R_{\mu\nu} + 1/2g_{\mu\nu} R = 0 \ R_{\mu$ $d\delta^{2} + u - GM/A^{2} - 3GMu^{2} = 0 \delta = 2GM/R \int \psi^{*} \psi dx = 1 P(x, t) = |\psi(x, t)|^{2} \psi(x, t) = |\psi(t)\rangle \langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \geq \hbar/2 \ p_{4} = -i\hbar\partial_{4} E = i\hbar\partial/\partial t \ H = p_{4} = -i\hbar\partial_{4} E = i\hbar\partial_{4} E$

D.P. George (George Robotics Ltd)

MicroPython on LEON

It's all about the RAM

If you ask me 'why is it done that way?', I will most likely answer: 'to minimise RAM usage'.

- Interned strings, most already in ROM.
- Small integers stuffed in a pointer.
- Optimised method calls (thanks PyPy!).
- Range object is optimised (if possible).
- Python stack frames live on the C stack.
- ROM absolutely everything that can be ROMed! $F_{\mu\sigma} \ \partial_{\nu} F^{\mu\nu} = J^{\nu} \ \partial_{\nu} \bar{F}^{\mu\nu} = 0 \ ds^2 = c^2 dt^2 - dx^2 - dy^2 - dz^2 \ ds^2 = g_{\mu\nu} dx^{\mu} dx^{\nu} dx^{\nu}$
- Garbage collection only (no reference counts).
- Exceptions implemented with custom set jmp/longjmp. $-\max \mathbf{F} = GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = q \left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right) - \hbar^2/2m\nabla^2 \psi$

 $v/e \cosh \xi = \gamma L = L_0/\gamma T = \gamma T_0 u' = (u - v)/(1 - uv/e^2) p = \gamma mv E = \gamma me^2 E^2 = p^2 e^2 + m^2 e^4 \partial_\mu J^\mu = 0 E_4 = -1/e\partial A_4/2 E_4 = -1/e^2 E$ $\mathrm{d}z^2 \mathrm{d}z^2 = g_{\mu\nu}\mathrm{d}z^\mu\mathrm{d}z^\nu \ g_{\mu\nu}(\mathrm{d}z^\mu/\mathrm{d}\tau)(\mathrm{d}z^\nu/\mathrm{d}\tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \ \mathrm{d}v^\mu/\mathrm{d}s + \Gamma^\mu_{\mu\sigma}v^\nu v^\sigma = 0 \ R^\mu_{\nu\rho\sigma} = \Gamma^\mu_{\nu\sigma,\rho} - \Gamma^\mu_{\nu\rho,\sigma} + \Gamma^\sigma_{\nu\rho,\sigma} + \Gamma^\sigma_{\nu\sigma,\mu} = \Gamma^\mu_{\nu\sigma,\nu} - \Gamma^\sigma_{\nu\sigma,\nu} + \Gamma^\sigma$ $\delta^{2} - r^{2} \sin^{2} \theta d\phi^{2} \Delta v \approx v_{s} GM(1/r_{s} - 1/r_{s}) d^{2}u/d\phi^{2} + u - GM/A^{2} - 3GMu^{2} = 0 \delta = 2GM/R \left[\psi^{*}\psi dx = 1 P(x, t) = |\psi(x, t)|^{2} \psi(x, t) = |\psi(x,$ $\mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B} / \partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E} / \partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x}) \psi(\mathbf{x}, t) = E \psi(\mathbf{x}, t) \mathbf{x}' = \gamma(\mathbf{x} - \mathbf{v}t) \mathbf{t}' = \gamma(t - \mathbf{v}t) \mathbf{x}' =$ $wv/c^2) \ p = \gamma mv \ E = \gamma mc^2 \ E^2 = p^2 c^2 + m^2 c^4 \ \partial_\mu J^\mu = 0 \ E_i = -1/c \partial A_i / \partial t - \partial_i \phi \ B_i = \epsilon_{ijk} \partial_j A_k \ F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu \ \bar{F}^{\mu\nu} = 1/2 \epsilon^{\mu i} / 2 \epsilon^{\mu i}$ $g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu} - g_{\nu\sigma,\mu} \right) dv^{\mu}/ds + \Gamma^{\mu}_{\nu\sigma} v^{\nu} v^{\sigma} = 0 \ R^{\mu}_{\mu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\sigma} \ R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} \ R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2 g_{\mu\nu} R = 0$

 $^{2} + u = GM/A^{2} = 3GMu^{2} = 0 \ \delta = 2GM/R \ \int \psi^{*} \psi dx = 1 \ P(x, t) = |\psi(x, t)|^{2} \ \psi(x, t) = |\psi(t)\rangle \ \langle x \rangle = \langle \psi | x | \psi \rangle \ \Delta x \Delta p \ge h/2 \ p_{4} = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = p_{4} \ A = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{4} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial/\partial$ D.P. George (George Robotics Ltd)

MicroPython on LEON

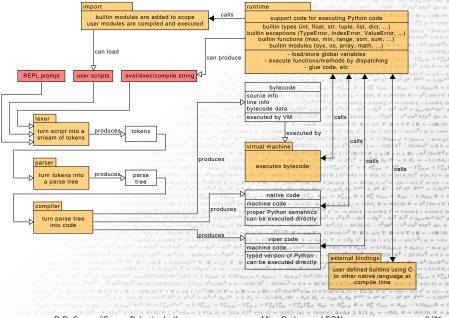
 $\mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E} / \partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi$

 $2GM/r)\mathrm{d}t^2 - \mathrm{d}r^2/(1 - 2GM/r) - r^2\mathrm{d}\theta^2 - r^2\sin^2\theta\mathrm{d}\phi^2 \Delta\nu \approx$ $\mathbf{F} = m\mathbf{a} \, \mathbf{F} = GMmr/r^3 \, \nabla \cdot \mathbf{E} = \rho/\epsilon \, \nabla \cdot \mathbf{B} = 0 \, \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

 $g_{\mu\nu}(\mathrm{d}x^{\mu}/\mathrm{d}\tau)(\mathrm{d}x^{\nu}/\mathrm{d}\tau) = 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$ $\partial B/\partial t \nabla \times B = \mu J + 1/c^2 \partial E/\partial t F = g (E + \mathbf{v} \times B) - \hbar^2/2m\nabla^2 \psi$

 $E = \gamma m c^2 E^2 = p^2 c^2 + m^2 c^4 \partial_\mu J^\mu = 0 E_i = -1/c \partial A_i$

Internals



D.P. George (George Robotics Ltd)

MicroPython on LEON

Object representation

A MicroPython object is a machine word, and has 3 different forms

Integers:

- Transparent transition to arbitrary precision integers.

Strings:

- Certain strings are not interned Objects: $^{2}\partial E/\partial t F = a(E + \mathbf{v} \times B) - h^{2}/2m\nabla^{2}\psi(\mathbf{x}, t) + V(\mathbf{x})\psi(\mathbf{x}, t) = E\psi(\mathbf{x}, t) \mathbf{x}' = \gamma(\mathbf{x} - \mathbf{v}t) \mathbf{t}' = \gamma(t - \mathbf{v})$
 - $\partial_{\mu}J^{\mu} = 0 E_i = -1/c\partial A_i/\partial t \partial_i \phi B_i = \epsilon_{ijk}\partial_j A_k F^{\mu\nu} = \partial^{\mu}A^{\nu} \partial^{\nu}A^{\mu} \tilde{F}^{\mu\nu} = 1/2\epsilon^{\mu}$ $-\Gamma^{\alpha}_{\nu\rho}\Gamma^{\mu}_{\alpha\sigma} R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} R = R^{\mu}_{\mu} G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$ $\langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \ge \hbar/2 \ p_i = -i\hbar \partial_i \ E = i\hbar \partial/\partial t \ H = p^2/2m + V$ $\gamma(x-vt) t' = \gamma(t-vx/c^2) \gamma = 1/(1-v^2/c^2)^{1/2} x' = -t \sinh \xi + x \cosh \xi t' = t \cosh \xi - x \sinh \xi$ to
 - A pointer to a structure.
 - $v^2/2m + V H|a\rangle = E|a\rangle U = e^{Ht/\hbar\hbar} \mathbf{F} = ma\mathbf{F} = GMmr/r^3 \nabla \cdot \mathbf{E} = a/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$ First element is a pointer to a type object.
 - ROMable (type, tuple, dictionary, function, module, ...). $/\mathrm{d}\tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \ \mathrm{d}v^{\mu}/\mathrm{d}s + \Gamma^{\mu}_{\nu\sigma}v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\mu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma}v^{\sigma} + \Gamma^{\alpha}_{\nu}v^{\sigma} + \Gamma^{\alpha}_{\nu}v^{\sigma} + \Gamma^{\alpha}_{\nu}v^{\sigma} + \Gamma^{\alpha}_$

Work on LEON port added representation for 64-bit NaN boxing. $\psi(x, t) = |\psi(t)\rangle \langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \ge \hbar/2 p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t$

D.P. George (George Robotics Ltd)

MicroPython on LEON

 $\frac{2}{2} \phi d\phi^2 \Delta \psi \approx \psi_1 GM(1/r_1 - 1/r_2) d^2 u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \delta = 2GM/R \left[\psi^* \psi dx = 1 P(x, t) = |\psi(x, t)|^2 \psi(x, t) = |\psi(x, t)|^2 \psi($

 $B = \mu J + 1/c^2 \partial E / \partial t F = q (E + v \times B) - \hbar^2 / 2m \nabla^2 \psi$

 $= e^{Ht/4\hbar}$ F = ma F = $GMmr/r^3$ $\nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0$ $\nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

 $\nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = g (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2/2m\nabla^2 \psi$ ²) $p = \gamma m v E = \gamma m c^2 E^2 = p^2 c^2 + m^2 c^4 \partial_\mu J^\mu = 0 E_i = -1/c \partial A_i$

 $^{\nu\rho\sigma}F_{\rho\sigma} \partial_{\nu}F^{\mu\nu} = J^{\nu} \partial_{\nu}\tilde{F}^{\mu\nu} = 0 ds^{2} = c^{2}dt^{2} - dx^{2} - dy^{2} - dz^{2} ds^{2} = g_{\mu\nu}dx^{\mu}dx^{\nu}dx^{\nu}dx^{\nu}dx^{\mu}dx^{\nu}dx^{\mu}dx^$ $G_{\mu\nu} = 8\pi G T_{\mu\nu} \, ds^2 = (1 - 2GM/r) dt^2 - dr^2/(1 - 2GM/r) - r^2 d\theta^2 - r^2 \sin^2\theta d\phi^2 \, \Delta\nu \approx$

 $g_{\mu\nu}(dx^{\mu}/d\tau)(dx^{\nu}/d\tau) = 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$

GitHub and the open-source community

https://github.com/micropython

MicroPython is a *public* project on GitHub.

- A global coding conversation.
- Anyone can clone the code, make a fork, submit issues, make pull requests
- MicroPython has over 2900 "stars" (top 0.02%), and more than 580 fork
- Contributions come from many people, with many different systems
- Leads to: more robust code and build system, more features, more $ds^2 = g_{\mu\nu} dx^\mu dx^\nu g$ $0 \ G_{\mu\nu} = 8\pi G T_{\mu\nu} \ ds^2 = (1 - 2GM/r)dt^2 - dr^2/(1 - 2GM/r) - r^2 d\theta^2 - r^2 \sin^2 \theta d\phi^2 \ \Delta\nu \approx 1 - 2GM/r + 2GM/$ supported hardware. $2m + V H|a\rangle = E|a\rangle U = e^{Ht/i\hbar} \mathbf{F} = m\mathbf{a}\mathbf{F} = GMm\mathbf{r}/r^3 \nabla \cdot \mathbf{E} = a/\epsilon \nabla$ $-x \sinh f \tanh f = v/c \cosh f = \gamma L = L_0/\gamma T = \gamma T_0 u' = (u - v)/(1 - uv/c^2) v$
- Hard to balance inviting atmosphere with strict code control.

 $c^2 + m^2 c^4 \partial_{\mu} J^{\mu} = 0 E_{\tau} = -1/c \partial A_{\tau}$ A big project needs many contributors, and open-source allows such projects to exist. $- p^2 c^2 + m^2 c^4 \partial_{\mu} J^{\mu} = 0 E_i = -1/c \partial A_i / \partial t - \partial_i \phi B_i = \epsilon_{ijk} \partial_i A_k F^{\mu\nu} = \partial^{\mu} A^{\nu} - \partial_i \phi B_i = \epsilon_{ijk} \partial_i A_k F^{\mu\nu} = \partial^{\mu} A^{\nu} - \partial_i \phi B_i = \delta_{ijk} \partial_i A_k F^{\mu\nu} = \partial_i \phi B_i = \delta_{ijk} \partial_i A_k F^{$ $+ \Gamma^{\mu}_{\mu\sigma} v^{\nu} v^{\sigma} = 0 \ R^{\mu}_{\mu\sigma\sigma} = \Gamma^{\mu}_{\mu\sigma,\sigma} - \Gamma^{\mu}_{\mu\sigma,\sigma} + \Gamma^{\alpha}_{\alpha\sigma} \Gamma^{\mu}_{\mu\sigma} - \Gamma^{\alpha}_{\alpha\sigma} \Gamma^{\mu}_{\alpha\sigma} R_{\mu\nu} = R^{\rho}_{\mu\nu\sigma} R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu}$

> $2GM/R \int \psi^* \psi dx = 1 P(x, t) = |\psi(x, t)|^2 |\psi(x, t) = |\psi(t)\rangle \langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \ge \hbar/2 p_i = -i\hbar\partial_i E = i\hbar\partial/\partial t$ D.P. George (George Robotics Ltd)

MicroPython on LEON 11/21

 $= R^{P}_{\mu\nu\rho} R = R^{P}_{\mu} G_{\mu\nu} = R_{\mu\nu} - 1/2q$

 $- \max \mathbf{F} = GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

Porting MicroPython to LEON

Small activity (ESTEC contract 4000114080) for 'MicroPython in Space'.

Objectives:

- Prototype port of MicroPython to LEON 2, on top of RTEMS 4.8
- 10 months duration, exploration of idea (no qualification)
- Special attention to the needs of Space:
 - determinism
 - concurrency
 - Iow use of resources (CPU, RAM)
 - interfacing to native (C/Ada) code

Deliverables:

- improved MicroPython core
- SPARC / LEON support
- RTEMS port with rtems module $\frac{1}{2} \int_{-1}^{2} \int_{$
- OBCP prototype engine $-\frac{dr^2}{dr} = \frac{dr^2}{dr} \frac{dr^2}{dr} = \frac{dr^2}{dr} + \frac{dr^2}{dr} = \frac{dr^2}{dr} + \frac{d$
- test suite
- reports: analysis, description of system, user manual

D.P. George (George Robotics Ltd)

MicroPython on LEON 12/21

 $^{2}\partial E_{i}/\partial I = q (E + v + B) - \hbar^{2}/2m\nabla^{2}\psi(x, t) + V(x)\psi(x, t) = E\psi(x, t) s' - \gamma(x - vt) t' - \gamma(t - v)$ $n^{2}e^{-\delta}\partial_{\mu}J^{\mu} = 0 = E_{i} - 1/c\partial A_{i}/\partial t - \partial_{i}\phi = B_{i} - e_{ij}k\partial_{j}A_{k} F^{\mu\nu} = \partial^{\mu}A^{\nu} - \partial^{\nu}A^{\mu} = h^{\mu\nu} = 1/2t^{\mu}$ $n^{\mu} = \Gamma^{\mu}_{\mu\nu,\mu} - \Gamma^{\mu}_{\mu\nu,\mu} + \Gamma^{\mu}_{\mu\nu} \Gamma^{\mu}_{\mu\nu} = \Gamma^{\mu}_{\mu\nu}R_{\mu} = R^{\mu}_{\mu\nu}R = R^{\mu}_{\mu}G_{\mu\nu} = R_{\mu\nu} - R^{\mu}_{\mu\nu}R_{\mu\nu}R_{\mu\nu}$

 $^{\mu}/\mathrm{d}\tau)(\mathrm{d}s^{\nu}/\mathrm{d}\tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \ \mathrm{d}s^{\mu}/\mathrm{d}s + \Gamma^{\mu}_{\mu\sigma}v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\mu\rho\sigma} = \Gamma^{\mu}_{\mu\rho,\sigma} - \Gamma^{\mu}_{\mu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma}$

 $\frac{2}{2} \phi d\phi^2 \Delta \psi \approx \psi_1 GM(1/r_1 - 1/r_2) d^2 u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \delta = 2GM/R \left[\psi^* \psi dx = 1 P(x, t) = |\psi(x, t)|^2 \psi(x, t) = |\psi(x, t)|^2 \psi($

 $b \|^2 \psi(x,t) = |\psi(t)\rangle \langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \ge \hbar/2 \ p_i = -i\hbar \partial_i \ E = i\hbar \partial/\partial t \ H = p^2/2m + V$ $b \| t' = \gamma (t - vx/c^2) \ \gamma = 1/(1 - v^2/c^2)^{1/2} \ x' = -t \sinh \xi + x \cosh \xi \ t' = t \cosh \xi - x \sinh \xi t$

 $1/2\epsilon^{\mu\nu\rho\sigma}F_{\rho\sigma}\partial_{\nu}F^{\mu\nu} = J^{\nu}\partial_{\nu}\tilde{F}^{\mu\nu} = 0 ds^2 = e^2 dt^2 - dx^2 - dy^2 - dz^2 dz^2 = g_{\mu\nu}dx^{\mu}dx^{\mu}dx^{\nu}s$ $G_{\mu\nu} = 8\pi GT_{\mu\nu}ds^2 = (1 - 2GM/r)dt^2 - dr^2/(1 - 2GM/r) - r^2 d\theta^2 - r^2 \sin^2\theta d\phi^2 \Delta\nu \approx$

 $\mathbf{F} = m\mathbf{a} \, \mathbf{F} = GMmr/r^3 \, \nabla \cdot \mathbf{E} = \rho/\epsilon \, \nabla \cdot \mathbf{B} = 0 \, \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

 $g_{\mu\nu}(dx^{\mu}/d\tau)(dx^{\nu}/d\tau) = 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$

The port to LEON: improvements to core

- separation of VM and compiler (only VM would need to be $E + y \times B = \hbar^2/2m\nabla^2 y$ qualified, about 200kB compiled excl. RTEMS)
- MicroPython cross compiler and persistent bytecode generation
- option for 64-bit NaN-boxing object model:
 - floats are objects: heap needed, good overall speed
 - floats are boxed: no heap (deterministic), faster FP ops, slower $/d\tau$) $(dx^{\nu}/d\tau) = 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu})$ overall due to 64-bit copying $= -\partial B/\partial t \nabla \times B = \mu J + 1/c^2 \partial E/\partial t F = g(E + \mathbf{v} \times B) - \hbar^2/2m\nabla^2 \psi$
- tool to order static hash tables to proper hashes
- understanding of determinism:
 - execution time of VM opcodes
 - allocation of heap memory
- optimisations to eliminate heap usage in places (eg iterators) $h \mu \approx \mu_1 GM(1/r_1 - 1/r_2) d^2 \mu/d\phi^2 + \mu - GM/A^2 - 3GMu^2$
- many bug fixes and speed optimisations (eg combining bytecodes)

 $b^{2} = \Delta v \approx v_{c} GM(1/r_{c} - 1/r_{c}) d^{2}u/d\phi^{2} + u - GM/A^{2} - 3GMu^{2} = 0 \delta = 2GM/R \left[\psi^{*}\psi dx = 1 P(x, t) = |\psi(x, t)|\right]$ $B/\partial t \nabla \times B = \mu J + 1/\epsilon^2 \partial E/\partial t F = \epsilon (E + \mathbf{x} \times B) - \hbar^2/2m \nabla^2 \phi(\mathbf{x}, t) + V(\mathbf{x}) \phi(\mathbf{x}, t) = E \phi(\mathbf{x}, t) \mathbf{x}^t$ $= \gamma m c^2 E^2 = p^2 c^2 + m^2 c^4 \partial_\mu J^\mu = 0 E_i = -1/c \partial A_i / \partial t - \partial_i \phi B_i = \epsilon_{ijk} \partial_j A_k F^{\mu\nu} = \partial^\mu A^\nu - \partial_i \phi B_i = \epsilon_{ijk} \partial_j A_k F^{\mu\nu} = \partial^\mu A^\nu - \partial_i \phi B_i = -1/c \partial_i \phi B_i$ ${}_{\sigma,\mu}) dv^{\mu}/ds + \Gamma^{\mu}_{\mu\sigma} v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\mu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\sigma} \ R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} \ R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu} R = 0 \ R^{\mu}_{\mu\nu} = R^{\mu}_{\mu\nu} + 1/2g_{\mu\nu} R = 0 \ R^{\mu}_{\mu\nu} = 0$

 $= 0 \ \delta = 2GM/R \ \int \psi^* \psi dx = 1 \ P(x,t) = |\psi(x,t)|^2 \ \psi(x,t) = |\psi(t)\rangle \ \langle x \rangle = \langle \psi | x | \psi \rangle \ \Delta x \Delta p \ge \hbar/2 \ p_i = -i\hbar \partial_i \ E = i\hbar \partial/\partial t \ H = p_i \ A = -i\hbar \partial_i \ E = i\hbar \partial_i \ A = -i\hbar \partial_i \ E = i\hbar \partial_i \ A = -i\hbar \partial_i \ E = i\hbar \partial_i \ A = -i\hbar \partial_i \ E = i\hbar \partial_i \ A = -i\hbar \partial_i \ E = i\hbar \partial_i \ A = -i\hbar \partial_i \ E = i\hbar \partial_i \ A = -i\hbar \partial_i \ E = i\hbar \partial_i \ A = -i\hbar \partial_i \ A = -$ D.P. George (George Robotics Ltd)

MicroPython on LEON 13/21

 $\partial t F = \sigma (E + \mathbf{v} \times B) - \hbar^2 / 2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x})\psi(\mathbf{x}, t) = E\psi(\mathbf{x}, t) \mathbf{x}' = \gamma(\mathbf{x} - \mathbf{v}t) \mathbf{t}' = \gamma(t - \mathbf{v})$

 $+ \Gamma^{\alpha}_{\nu\sigma} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\sigma} R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} R = R^{\mu}_{\mu} G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$

 $t = |\psi(t)\rangle \langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \ge \hbar/2 p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t H = p^2/2m + V$ $(t - vx/c^2) = 1/(1 - v^2/c^2)^{1/2} x' = -t \sinh \xi + x \cosh \xi t' = t \cosh \xi - x \sinh \xi ta$ $F_{\mu\mu} \partial_{\nu} F^{\mu\nu} = J^{\nu} \partial_{\nu} \bar{F}^{\mu\nu} = 0 \, ds^2 = c^2 dt^2 - dx^2 - dy^2 - dz^2 \, ds^2 = g_{\mu\nu} dx^{\mu} dx^{\nu} g_{\mu\nu}$

 $GT_{\mu\nu} ds^2 = (1 - 2GM/r)dt^2 - dr^2/(1 - 2GM/r) - r^2 d\theta^2 - r^2 \sin^2 \theta d\phi^2 \Delta \nu \approx$ $Ht/i\hbar \mathbf{F} = m\mathbf{a} \mathbf{F} = GMm\mathbf{r}/r^3 \nabla \cdot \mathbf{E} = a/\epsilon \nabla \cdot \mathbf{B} = 0$

 $c^{2}E^{2} = v^{2}c^{2} + m^{2}c^{4}\partial_{\mu}J^{\mu} = 0 E_{\ell} = -1/c\partial A_{\ell}$

The port to LEON: LEON specifics

- support for SPARC v8 architecture
- $0 \delta = 2GM/R ||\psi^*\psi dx = 1 P(x, t) = |\psi(x, t)|^2 |\psi(x, t) = |\psi(x, t)|^2$ ability to have multiple VMs running in the same address space, each with their own heap
- multitasking delegated to RTEMS (atomicity of ops guaranteed)
- synchronisation achieved with RTEMS queues
- multitasking also available via Python co-routines using "yield", $\mathbf{B} \mathbf{I} \partial (\mathbf{P} - \mathbf{r} (\mathbf{D} \mathbf{V} \times \mathbf{B}) - \hbar^2/2m\nabla^2 \psi$ $^{\sigma} = 0 R^{\mu}_{\nu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma}$ good for soft real-time applications (can share heap)
- creation of the rtems module (queue, task, semaphore, timer) -
- $t' = -t \sinh \xi + x \cosh \xi$ $t' = t \cosh \xi x \sinh \xi$ to datapool module to share data (from C and Python)
- $E|a\rangle U = e^{Ht/i\hbar}$ F = ma F = $GMmr/r^3 \nabla \cdot E = a/\epsilon \nabla \cdot B = 0 \nabla \times E = -\partial B/\partial t$

 $\frac{1}{2} + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \psi^* \psi dx = 1 \ P(x,t) = |\psi(x,t)|^2 \ \psi(x,t) = |\psi(t)\rangle \ \langle x \rangle = \langle \psi | x | \psi \rangle \ \Delta x \Delta p \ge \hbar/2 \ p_4 = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = p_4 \ A = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = -i\hbar\partial_4 \ E = -i\hbar\partial_4 \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = -i\hbar\partial_4 \ E = -i\hbar\partial_4 \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_4 \ E = -i\hbar\partial$ D.P. George (George Robotics Ltd)

MicroPython on LEON 14/21

 $\mathbf{F} = \max \mathbf{F} = GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = q \left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right) - \hbar^2/2m\nabla^2 \psi$ $v/c \cosh \xi = \gamma \ L = L_0 / \gamma \ T = \gamma T_0 \ u' = (u - v) / (1 - uv/c^2) \ p = \gamma mv \ E = \gamma mc^2 \ E^2 = p^2 c^2 + m^2 c^4 \ \partial_\mu J^\mu = 0 \ E_i = -1/c \partial A_i / c^2 + c$ $= y_{\mu\nu} \mathrm{d}x^{\mu} \mathrm{d}x^{\nu} \ y_{\mu\nu} (\mathrm{d}x^{\mu}/\mathrm{d}\tau) (\mathrm{d}x^{\mu}/\mathrm{d}\tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (y_{\mu\nu,\sigma} + y_{\mu\sigma,\nu} - y_{\nu\sigma,\mu}) \ \mathrm{d}v^{\mu}/\mathrm{d}s + \Gamma^{\mu}_{\nu\sigma} v^{\nu} v^{\sigma} = 0 \ R^{\mu}_{\nu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\sigma,\rho} + \Gamma^{\alpha}_{\nu\sigma} + \Gamma^{\alpha}_{\sigma} + \Gamma^{\alpha}_{\sigma$ $\theta^2 - r^2 \sin^2 \theta d\phi^2 \Delta \nu \approx \nu_t GM(1/r_t - 1/r_t) d^2 u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \delta = 2GM/R \left[\psi^* \psi dx = 1 P(x, t) = |\psi(x, t)|^2 \psi(x, t) = |\psi(x, t)$ $\mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B} / \partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E} / \partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x}) \psi(\mathbf{x}, t) = E \psi(\mathbf{x}, t) \mathbf{x}' = \gamma(\mathbf{x} - \mathbf{v}t) \mathbf{t}' = \gamma(t - \mathbf{v}t) \mathbf{x}' =$ $\bar{\tau}_{\mu\sigma,\nu} - \bar{\tau}_{\nu\sigma,\mu} \right) dv^{\mu}/ds + \Gamma^{\mu}_{\mu\sigma} v^{\nu}v^{\sigma} = 0 \ R^{\nu}_{\mu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\sigma} R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0 \ R^{\mu}_{\mu\nu} = 0 \ R^{\mu}_{\mu\nu\rho} R_{\mu\nu} = 0 \ R^{\mu}_{\mu\nu} R_{\mu\nu} = 0 \ R^{\mu}_{\mu\nu\rho} R_{\mu\nu} = 0 \ R^{\mu}_{\mu\nu} R_{\mu\nu} R_{\mu\nu} = 0 \ R^{\mu}_{\mu\nu} R_{\mu\nu} R_{$

 $r/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

 $\mu J + 1/c^2 \partial E / \partial t F = q (E + v \times B) - \hbar^2 / 2m \nabla^2 \psi$

The port to LEON

Heap management:

- traditional problem, it can be non-deterministic
- IBM metronome GC is too complex for qualification
- simplest solution: allocate beforehand, most functionality uses stack
- heap_lock and heap_unlock methods (exception raised on OB/Ot allocation, can be managed) $\mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = g (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2/2m\nabla^2 \psi$ $(c^2) = -2mv E = -2mc^2 E^2 = v^2 c^2 + m^2 c^4 \partial_{\mu} J^{\mu} = 0 E_{\mu} = -1/c \partial A_{\mu}$

Performance:

- $\Gamma^{\mu}_{\nu\sigma,\rho} \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma}\Gamma^{\mu}_{\alpha\rho} \Gamma^{\alpha}_{\nu\rho}\Gamma^{\mu}_{\alpha\sigma} R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} R = R^{\mu}_{\mu} G_{\mu\nu} = R_{\mu\nu} 1/2g_{\mu\nu}R = 0$ about 100x slower than equivalent C code (expected, similar to PC)
- implement performance critical code in C and wrap it (easy to do)
- Python is an application-level language, fast development $-3GMu^2 = 0 \delta =$ = ma $\mathbf{F} = GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F}$ $= a(\mathbf{E} + \mathbf{x} \times \mathbf{B}) - b^2/2m\nabla^2 \mathbf{x}$

 $= v/c \cosh t = \gamma L = L_0/\gamma T = \gamma T_0 u' = (u - v)/(1 - uv/c^2) p = \gamma m v E = \gamma m c^2 E^2 = p^2 c^2 + m^2 c^4 \partial_{\mu} J^{\mu} = 0 E_c = -1/c \partial A_c$ $=g_{\mu\nu}dx^{\mu}dx^{\nu}\ g_{\mu\nu}(dx^{\mu}/d\tau)(dx^{\nu}/d\tau) = 1\ \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})\ dv^{\mu}/dx + \Gamma^{\mu}_{\mu\sigma}v^{\nu}v^{\sigma} = 0\ R^{\mu}_{\mu\sigma,\sigma} - \Gamma^{\mu}_{\mu\sigma,\sigma} + \Gamma^{\mu}_{\mu\sigma,\sigma} +$ $=r^{2} \sin^{2} \theta d\phi^{2} \ \Delta \nu \approx \nu_{i} GM(1/r_{i}-1/r_{f}) \ d^{2}u/d\phi^{2} + u - GM/A^{2} - 3GMu^{2} = 0 \ \delta = 2GM/R \ \int \psi^{*} \psi \, dx = 1 \ P(x,t) = |\psi(x,t)|^{2} \ \psi(x,t) = |\psi$ $\nabla \mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \nabla \mathbf{B} = u\mathbf{I} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = a(\mathbf{E} + \mathbf{v} \times \mathbf{B}) - b^2/2m\nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{v})\psi(\mathbf{x}, t) = E\psi(\mathbf{x}, t) \mathbf{x}' = a(\mathbf{x} - \mathbf{v}) \mathbf{t}' = a(t - \mathbf{v}) \mathbf{x}' + b^2 (t - \mathbf$ $=\gamma m v E = \gamma m c^2 E^2 = p^2 c^2 + m^2 c^4 \partial_\mu J^\mu = 0 E_i = -1/c \partial A_i / \partial t - \partial_i \phi B_i = \epsilon_{ijk} \partial_j A_k F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu \tilde{F}^{\mu\nu} = 1/2 \epsilon^{\mu i} - 1/2 \epsilon^{\mu i} + 1/2 \epsilon^{$ $g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu} \right) dv^{\mu}/ds + \Gamma^{\mu}_{\nu\sigma}v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\nu\sigma,\sigma} - \Gamma^{\mu}_{\nu\sigma,\sigma} + \Gamma^{\alpha}_{\nu\sigma,\sigma} \Gamma^{\mu}_{\mu\sigma} - \Gamma^{\alpha}_{\mu\sigma}\Gamma^{\mu}_{\mu\sigma} R_{\mu\nu} = R^{\mu}_{\mu\nu\sigma,\rho} R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$

 $2^{2} - 3GMu^{2} = 0 \ \delta = 2GM/R \ \int \psi^{*} \psi dx = 1 \ P(x, t) = |\psi(x, t)|^{2} \ \psi(x, t) = |\psi(t)\rangle \ \langle x \rangle = \langle \psi | x | \psi \rangle \ \Delta x \Delta p \ge \hbar/2 \ p_{i} = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = p_{i} \ A = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_{i} \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial/\partial$ D.P. George (George Robotics Ltd)

MicroPython on LEON 15/21

 $\partial \mathbf{E} / \partial t \mathbf{F} = \sigma (\mathbf{E} + \mathbf{y} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x}) \psi(\mathbf{x}, t) = E \psi(\mathbf{x}, t) \mathbf{x}' = \gamma (\mathbf{x} - \mathbf{v}t) \mathbf{t}' = \gamma (t - \mathbf{v})$

 $= GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

 $= \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2/2m\nabla^2 \psi$

OBCP prototype engine

A prototype On-Board Control Procedure engine, not qualified, just for demonstration purposes.

Features:

- 4 VM instances (2 idle, 2 running)
- VM interface: load, execute, pause, resume, step, stop
- simulated Ground sends precompiled bytecode
- C tasks, Python tasks running together
- communication via queues and datapool $\frac{1}{2} \int_{0}^{2\pi} \int_{0}^{$ $(t - vx/c^2) = 1/(1 - v^2/c^2)^{1/2} x' = -t \sinh \xi + x \cosh \xi t' = t \cosh \xi - x \sinh \xi ta$
- ${}^{\nu}A^{\mu} \ \dot{F}^{\mu\nu} = 1/2 \epsilon^{\mu\nu\rho\sigma} F_{\rho\sigma} \ \partial_{\nu}F^{\mu\nu} = J^{\nu} \ \partial_{\nu} \ddot{F}^{\mu\nu} = 0 \ \mathrm{d} s^{2} = c^{2} \mathrm{d} t^{2} \mathrm{d} x^{2} \mathrm{d} x^{2} \mathrm{d} x^{2} \ \mathrm{d} s^{2} = g_{\mu\nu} \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \ \mathrm{d} s^{\mu} \mathrm{d$ demonstrates calls to native code
- heap is locked after start-up $dx^2 dx^2 dx^2 + dx^2 = g_{\mu\nu}dx^{\mu}dx^{\nu}g_{\mu\nu}(dx^{\mu}/dr)(dx^{\nu}/dr) = 1$ $\Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\mu} g_{\nu\sigma,\mu})$

 $\phi^2 + u = GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \psi^* \psi dx = 1 \ P(x,t) = |\psi(x,t)|^2 \ \psi(x,t) = |\psi(t)\rangle \ \langle x \rangle = \langle \psi | x | \psi \rangle \ \Delta x \Delta p \ge \hbar/2 \ p_i = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = p_i \ A = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial_i \ E = -i\hbar\partial/\partial t \ H = -i\hbar\partial/\partial$ D.P. George (George Robotics Ltd)

MicroPython on LEON 16/21

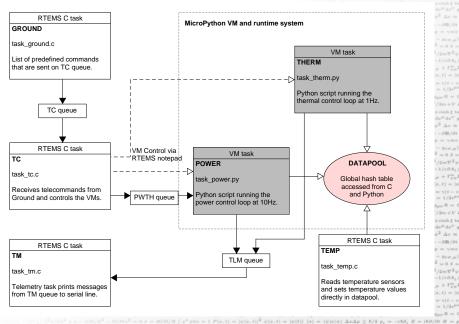
 $+V H|a\rangle = E|a\rangle U = e^{Ht/i\hbar} F = maF = GMmr/r^3 \nabla \cdot E = a/\epsilon \nabla \cdot B = 0 \nabla \times E = -\partial B/\partial t$

 $U = e^{Ht/\hbar} \quad F = \max F = GMmr/r^3 \nabla \cdot E = \rho/\epsilon \nabla \cdot B = 0 \nabla \times E = -\partial B/\partial t \nabla \times B = \mu J + 1/c^2 \partial E/\partial t F = \rho(E + y \times B) - \hbar^2/2m\nabla^2 \psi$ $dt \xi = v/c \cosh \xi = \gamma L = L_0/\gamma T = \gamma T_0 u' = (u - v)/(1 - uv/c^2) p = \gamma mv E = \gamma mc^2 E^2 = p^2 c^2 + m^2 c^4 \partial_{\mu} J^{\mu} = 0 E_4 = -1/c\partial A_4/c^4 + 2 c^4 + 2$ $\mathrm{d} y^2 - \mathrm{d} z^2 - \mathrm{d} z^2 - y_{\mu\nu} \mathrm{d} z^\mu \mathrm{d} z^\nu - y_{\mu\nu} \mathrm{d} z^\mu \mathrm{d} \tau) (\mathrm{d} z^\nu / \mathrm{d} \tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \ \mathrm{d} v^\mu / \mathrm{d} s + \Gamma^\mu_{\mu\sigma} v^\nu v^\sigma = 0 \ R^\mu_{\mu\sigma,\rho} - \Gamma^\mu_{\mu\sigma,\rho} - \Gamma^\mu_{\mu\sigma,\rho} + \Gamma^\alpha_{\mu\sigma} +$ $M/r) = r^2 d\theta^2 = r^2 \sin^2 \theta d\phi^2 \quad \Delta \nu \approx \nu_c G M (1/r_c - 1/r_c) d^2 u / d\phi^2 + u - G M / A^2 - 3G M u^2 = 0 \\ \delta = 2G M / R \left[\psi^* \psi dx = 1 P(x, t) = |\psi(x, t)|^2 \psi(x, t) = |\psi(x, t)|^2 \right] = 0$ $\nabla \cdot \mathbf{B} = 0 \ \nabla \times \mathbf{E} = -\partial \mathbf{B} / \partial t \ \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E} / \partial t \ \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x}) \psi(\mathbf{x}, t) = E \psi(\mathbf{x}, t) \ x' = \gamma(x - vt) \ t' = \gamma(t - v) + \frac{1}{2} (x - v)$ $\psi)/(1-\psi\nu/c^2) p = \gamma m\nu E = \gamma mc^2 E^2 = p^2c^2 + m^2c^4 \partial_{\mu}J^{\mu} = 0 E_i = -1/c\partial A_i/\partial t - \partial_i\phi B_i = \epsilon_{ijk}\partial_jA_k F^{\mu\nu} = \partial^{\mu}A^{\nu} - \partial^{\nu}A^{\mu} \tilde{F}^{\mu\nu} = 1/2\epsilon^{\mu} + 1/2\epsilon^{\mu} +$ $(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) dv^{\mu}/ds + \Gamma^{\mu}_{\mu\sigma} v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\nu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\sigma} \ R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} \ R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu} R = 0$

 $GMmr/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

 $(1 - 2GM/r)dt^2 - dr^2/(1 - 2GM/r) - r^2d\theta^2 - r^2\sin^2\theta d\phi^2 \Delta\nu \approx$ $\mathbf{F} = m\mathbf{a} \, \mathbf{F} = GMmr/r^3 \, \nabla \cdot \mathbf{E} = \rho/\epsilon \, \nabla \cdot \mathbf{B} = 0 \, \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

OBCP prototype engine



D.P. George (George Robotics Ltd)

MicroPython on LEON

Thermal script

$$\begin{split} & (x_1(x_1, x_1) = E^{i_1}(x_1), x' = i(x - \alpha t) t' = i(1 - \alpha x)t' \\ & (\beta + \alpha - 1) - i(A_A, F^{iii}) = 0^{i_1}A_A^{i_2} - 0^{i_1}A_A^{i_2}F^{i_2} = 1(2x^{i_1}) \\ & (\beta + 1) - i(1 - \alpha x)t' \\ & (\beta + 1) - i(1 -$$

4.0	$y = y_1 + y_2 + y_1 + y_2 + y_2 + y_2 + y_1 + y_2 + y_2 + y_1 + y_2 + y_2 + y_1 + y_2 + $	$\langle e^2 \rangle p = \gamma m v$
	main(): print('THERM script started, using VM', rtems.script_id())	$-g_{\nu\sigma,\mu})^{2} = 0 \delta =$
	# get the TH queue tm g = rtems.gueue.ident('TLMD')	$\frac{1}{2m\nabla^2\psi}$ - $\frac{1}{c\partial A_i}$
	de = datanol.ident'DHTPPDUL')	$\sigma + \Gamma^{\alpha}_{\nu\sigma} \mathbf{I}$ $(x, t) = \psi $
	up - uatapuot.luent (unintuu.) # create the arrays to hold the values and thresholds	$= \gamma (t - v)$ $= 1/2e^{\mu t}$
	temp_val = array.array('d', N_TEMP * (0)) temp_thresh = array.array('d', N_TEMP * (0, 0))	$g_{\mu\nu}R = 0$ /2m + V k
	# get initial thresholds from the datapool dp.get_buf(K_DP_TEMP_THRESH_38, temp_thresh)	$x \sinh \xi$ ta $dx^{\mu} dx^{\nu} g$
	# print initial thresholds for i in range(N-IEHP):	$\phi^2 \Delta \nu \approx -\partial \mathbf{B} / \partial t$
	<pre>print('THERM: initial temp_thres(%d): (%.2f, %.2f)' % (i, temp_thresh[2 * i], temp_thresh[2 * i * 1]))</pre>	$p = \gamma m v$ $- g_{\nu\sigma, \mu}$
	<pre># buffer for TM messages (will be populated using struct.pack_into) buf = bytearray(26)</pre>	$2^{2} = 0 \delta = \frac{1}{2m\nabla^{2}\psi}$
	# from now on we are deterministic micropython.heap_lock()	$-1/c\partial A_i/a$ $\sigma + \Gamma_{\mu\sigma}^{\alpha}$
	# control loop runs at 1Hz while True:	$(x, t) = \psi $
	H get Lenperatures from the delapool dp.get_buf(K_DP_TEHP_VHL_30, temp_val)	$= \gamma(t - v)$ = 1/2e ^{µ1}
	# pet thresholds from the datapool dp.get_buf(K_DP_TEMP_THRESH_30, temp_thresh)	$g_{\mu\nu}R = 0$ /2m + V/k $x \sinh \xi$ ta
	# check thresholds for i in range(N_IEMP):	$dx^{\mu} dx^{\nu} g$ $\phi^2 \Delta \nu \approx$
	<pre>if not temp_thresh[2 * i] <= temp_val[i] <= temp_thresh[2 * i + 1]: struct.pack_into(">BOOdd", buf, 0, K_TM_TEMP_RANGE, i,</pre>	$-\partial \mathbf{B} / \partial t$
	temp_thresh(2 * i], temp_thresh(2 * i + 1], temp_val(i)) tm_q.send(buf, rtems.HAHI)	$p = \gamma m v$ $- g_{\nu\sigma,\mu}$
90	$\frac{15.8-1}{16} = E = e^{Ht/th} \mathbf{F} = \max \mathbf{F} = GMmr/c^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$	$73z^2 = 0 \delta =$
	$1 + (\sinh \xi - x \sinh \xi - v/c \cosh \xi - \gamma L - L_0/\gamma T - \gamma T_0 u' = (u - v)/(1 - vv/c^2) p - mv v E - mv^2 E^2 - p^2 c^3 + m^2 c^4 \partial_\mu J^\mu = 0$ $1 + (1 - v)/c^2 + dy^2 - dy^2 - dy^2 - dy^2 - dy^2 + dy^2 - dy^$	$\Sigma_i = -1/c\partial A_i/c$
	$(\pi^2 + w^2)(1 + 2GM/r) - r^2 d\theta^2 - r^2 \sin^2 \theta d\phi^2 \ \Delta v \approx v_i GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = \psi(x, t) + 2GM(1/r_i - 1/r_f) \ d^2u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \phi^* \psi dx = 1 \ P(x, t) = 0 \ (x, t) $	$ ^2 \psi(x, t) = \psi $
	$\begin{split} &\cos \mathbb{P} = GMmt/\delta^3 \ \nabla \cdot \mathbb{E} = \rho/\epsilon \ \nabla \cdot \mathbb{B} = 0 \ \nabla \times \mathbb{E} = -\partial \mathbb{B}/\partial t \ \nabla \times \mathbb{B} = \rho \mathbf{J} + 1/c^2 \partial \mathbb{E}/\partial t \ \mathbb{F} = q \left(\mathbb{E} + \mathbf{v} \times \mathbb{B}\right) \\ &- \hbar^2/2m \ \nabla^2 \psi(\mathbf{x}, t) + \mathbf{V}(\mathbf{x}) \psi(\mathbf{x}, t) = \mathbb{E}\psi(\mathbf{x}, t) \ \mathbf{x}' = \gamma(\mathbf{x} - \mathbf{x}) \\ &+ (\nabla \cdot \mathbf{v}) (\nabla - \mathbf{v}) (1 - wv/c^2) \ p = \gamma mv \ \mathbb{E} = \gamma mc^2 \ \mathbb{E}^2 = p^2 c^2 + m^2 c^4 \ \partial_{\mu} J^{\mu} = 0 \ \mathbb{E}_i = -1/c \partial A_i / \partial t - \partial_i \phi \ B_i = \epsilon_{ijk} \partial_j A_k \ \mathbb{F}^{\mu\nu} = \partial^{\mu} A^{\nu} - \partial^{\nu} A^{\mu} \\ &- \partial^{\mu} A^{\mu} = 0 \ \mathbb{E}_i = -1/c \partial A_i / \partial t - \partial_i \phi \ B_i = \epsilon_{ijk} \partial_j A_k \ \mathbb{F}^{\mu\nu} = \partial^{\mu} A^{\nu} - \partial^{\nu} A^{\mu} \\ &- \partial^{\mu} A^{\mu} = 0 \ \mathbb{E}_i = -1/c \partial A_i / \partial t - \partial_i \phi \ B_i = \epsilon_{ijk} \partial_j A_k \ \mathbb{F}^{\mu\nu} = 0 \ \mathbb{E}_i + 2 (1 - wv/c^2) \ \mathbb{E}_i + 2 ($	$\tilde{F}^{\mu\nu} = 1/2\epsilon^{\mu\nu}$
	$\begin{split} (b) &= 1 \ \Gamma_{\mu\nu\sigma} = -1/2 (g_{\mu\nu\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \ dv^{\mu}/ds + \Gamma^{\mu}_{\mu\sigma} v^{\nu} \sigma^{\sigma} = 0 \ R^{\mu}_{\mu\sigma\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\mu,\sigma} + \Gamma^{\nu}_{\mu\sigma} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\mu\rho} \Gamma^{\mu}_{\alpha\sigma} - R^{\mu}_{\mu\nu\rho} R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} = R_{\mu\nu\rho} \ R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} = R_{\mu\nu\rho} \ R = R^{\mu}_{\mu\sigma} \ G_{\mu\nu} = R_{\mu\nu} = R_{\mu\nu\rho} \ R = R^{\mu}_{\mu\sigma} \ G_{\mu\nu} = R_{\mu\nu} = R_{\mu\nu\rho} \ R = R^{\mu}_{\mu\sigma} \ G_{\mu\nu} = R^{\mu}_{\mu\sigma} \ G_{\mu\nu} = R^{\mu}_{\mu\sigma} \ G_{\mu\nu} = R^{\mu}_{\mu\sigma} \ G_{\mu\nu} = R^{\mu}_{\mu\sigma} \ R^{\mu}_{\mu\sigma} \ R^{\mu}_{\mu\sigma} = R^{\mu}_{\mu\sigma} \ R^{\mu}_{\mu\sigma} \ R^{\mu}_{\mu\sigma} = R^{\mu}_$	$-1/2g_{\mu\nu}R = 0$ $i\hbar\partial/\partial t H = n$
	D.P. George (George Robotics Ltd) MicroPython on LEON	18/21

License and availability

- MicroPython core: MIT license
- the port to LEON: "ESA Community License Type 3, permissive" $1/(1 - v^2/c^2)^{1/2} x' = -t \sinh \xi + x \cosh \xi t' = t \cosh \xi - x \sinh \xi ta$ (restricted to ESA states)

 $\nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E} / \partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi$

 $_{\mu\nu}(dx^{\mu}/d\tau)(dx^{\nu}/d\tau) = 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$

19/21

- $|a\rangle = E|a\rangle U = e^{Ht/i\hbar} F = maF = GMmr/r^3 \nabla \cdot E = \rho/\epsilon \nabla \cdot B = 0 \nabla \times E = -\partial B/\partial t$ for exporting outside ESA region, talk to me
- code available (mid-end June) in European Space Software Repository: https://essr.esa.int/ × B) $-\hbar^2/2m\nabla^2\psi(\mathbf{x},t) + V(\mathbf{x})\psi(\mathbf{x},t) = E\psi(\mathbf{x},t) \mathbf{x}' = \gamma(\mathbf{x}-\mathbf{v}t) \mathbf{t}' = \gamma(t-\mathbf{v})$
- documents available: Analysis and Adaptation; Test Spec and Report, Executive Summary, Final Report, User Manual $= 0 \ G_{\mu\nu} = 8\pi G T_{\mu\nu} \ ds^2 = (1 - 2GM/r) dt^2 - dr^2/(1 - 2GM/r) - r^2 d\theta^2 - r^2 \sin^2 \theta d\phi^2 \ \Delta\nu \approx$

 $p_{1} = -i\hbar\partial_{1}E = i\hbar\partial/\partial t H = p^{2}/2m + V H|a\rangle = E|a\rangle U = e^{Ht/i\hbar} \mathbf{F} = m\mathbf{a} \mathbf{F} = GMm\mathbf{r}/r^{3} \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$ $J^{\nu} \ \partial_{\nu} f^{\mu\nu} = 0 \ \mathrm{d} s^2 = c^2 \mathrm{d} t^2 - \mathrm{d} s^2 - \mathrm{d} y^2 - \mathrm{d} z^2 \ \mathrm{d} s^2 = g_{\mu\nu} \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \ g_{\mu\nu} (\mathrm{d} x^{\mu}/\mathrm{d} \tau) (\mathrm{d} x^{\nu}/\mathrm{d} \tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \ \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \ \mathrm{d} x^{\mu}/\mathrm{d} \tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \ \mathrm{d} x^{\mu}/\mathrm{d} \tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \ \mathrm{d} x^{\mu}/\mathrm{d} \tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \mathrm{d} x^{\mu} \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \ \mathrm{d} x^{\mu}/\mathrm{d} \tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} \ \mathrm{d} x^{\mu}/\mathrm{d} \tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \mathrm{d} x^{\mu} \mathrm{d} x^{\mu} \mathrm{d} x^{\mu} \mathrm{d} x^{\mu} \mathrm{d} x^{\mu} \mathrm{d} x^{\mu}/\mathrm{d} \tau) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \mathrm{d} x^{\mu} \mathrm{d} x^{$ $E[a] U = e^{Ht/\hbar} F = ma F = GMmr/r^3 \nabla \cdot E = \rho/\epsilon \nabla \cdot B = 0 \nabla \times E = -\partial B/\partial t \nabla \times B = \mu J + 1/c^2 \partial E/\partial t F = g(E + y \times B) - \hbar^2/2m\nabla^2 \psi$ $\tanh \xi = v/c \cosh \xi = \gamma \ L = L_0/\gamma \ T = \gamma T_0 \ u' = (u-v)/(1-uv/c^2) \ p = \gamma mv \ E = \gamma mc^2 \ E^2 = p^2 c^2 + m^2 c^4 \ \partial_\mu J^\mu = 0 \ E_i = -1/c \partial A_i$ $\mathrm{d} x^2 - \mathrm{d} x^2 - g_{\mu\nu} \mathrm{d} x^{\mu} \mathrm{d} x^{\nu} - g_{\mu\nu} \mathrm{d} x^{\mu} \mathrm{d} r) (\mathrm{d} x^{\mu} / \mathrm{d} r) (\mathrm{d} x^{\nu} / \mathrm{d} r) = 1 \ \Gamma_{\mu\nu\sigma} = 1/2 (g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) \ \mathrm{d} v^{\mu} / \mathrm{d} s + \Gamma^{\mu}_{\mu\sigma} v^{\nu} v^{\sigma} = 0 \ R^{\mu}_{\nu\rho\sigma} - \Gamma^{\mu}_{\nu\rho,\sigma} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma} + \Gamma^{$ $= r^{2} d\theta^{2} - r^{2} \sin^{2} \theta d\phi^{2} \Delta y \approx y_{c} GM(1/r_{c} - 1/r_{c}) d^{2} y/d\phi^{2} + y - GM/A^{2} - 3GMy^{2} = 0 \delta = 2GM/R \left[\psi^{*}\psi dx = 1 P(x, t) = |\psi(x, t)|^{2} \psi(x, t) = |\psi(x, t)|^{2} \left[\psi(x, t) + \frac{1}{2} \psi(x, t) + \frac{1}{$

 $\nabla \cdot \mathbf{B} = 0 \ \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t \ \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \ \mathbf{F} = g \left(\mathbf{E} + \mathbf{v} \times \mathbf{B} \right) - \hbar^2 / 2m \nabla^2 \psi(\mathbf{x}, t) + V(\mathbf{x}) \psi(\mathbf{x}, t) = E\psi(\mathbf{x}, t) \ x' = \gamma(x - vt) \ t' = \gamma(t - v)$ $= uv/c^2) \ p = \gamma mv \ E = \gamma mv^2 \ E^2 = p^2 c^2 + m^2 c^4 \ \partial_\mu J^\mu = 0 \ E_i = -1/c \partial A_i / \partial t - \partial_i \phi \ B_i = \epsilon_{ijk} \partial_j A_k \ F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu \ \bar{F}^{\mu\nu} = 1/2 \epsilon^{\mu i} / 2 \epsilon^{\mu i$ $g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu} \right) dv^{\mu}/ds + \Gamma^{\mu}_{\nu\sigma} v^{\nu}v^{\sigma} = 0 \ R^{\mu}_{\nu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\nu\rho} \Gamma^{\mu}_{\alpha\sigma} \ R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} \ R = R^{\mu}_{\mu} \ G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu} R = 0$ $\phi^2 + u = GM/A^2 - 3GMu^2 = 0 \ \delta = 2GM/R \ \int \psi^* \psi dx = 1 \ P(x,t) = |\psi(x,t)|^2 \ \psi(x,t) = |\psi(t)\rangle \ \langle x \rangle = \langle \psi | x | \psi \rangle \ \Delta x \Delta p \ge \hbar/2 \ p_4 = -i\hbar\partial_4 \ E = i\hbar\partial/\partial t \ H = p_4 \ A^2 + i\hbar\partial/\partial t \ A^2 +$ D.P. George (George Robotics Ltd) MicroPython on LEON

Conclusions and Future activities

A powerful and modern language, large community, powerful tools now available for constrained/embedded systems!

Possible applications in Space: general purpose application language payloads, OBCP engine

Other applications:

- schools/teaching (micro:bit, pyboard, high-schools and universities) $2GM/r)dt^2 - dr^2/(1 - 2GM/r) - r^2 d\theta^2 - r^2 \sin^2 \theta d\phi^2$
- hobbyists and hackers
- $n = E = -2mc^2 E^2 = r^2 c^2 + m^2 c^4 \partial_{ij} J^{\mu} = 0 E_i = -2mc^2 E_i = -2mc^2$ embedded engineers, to make prototyping easier
- great potential for IoT

 $=\gamma(t-vx/c^2)$ $\gamma = 1/(1-v^2/c^2)^{1/2} x' = -t \sinh \xi + x \cosh \xi t' = t \cosh \xi - x \sinh \xi$ to Continued software/hardware development:

- Python 3.5 support, and improved compatibility with CPython
- partnering with chip vendors to support their MCUs $\sigma = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu}) dv^{\mu}/ds + \Gamma^{\mu}_{\nu\sigma}v^{\nu}v^{\sigma} = 0 R^{\mu}_{\nu\sigma,\sigma} = \Gamma^{\mu}_{\nu\sigma,\sigma} - \Gamma^{\mu}_{\nu\sigma,\sigma} + \Gamma^{\alpha}_{\nu\sigma}$
- development of new boards

D.P. George (George Robotics Ltd)

 $2GM/R \int \psi^* \psi dx = 1 P(x,t) = |\psi(x,t)|^2 |\psi(x,t) = |\psi(t)\rangle \langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta p \ge \hbar/2 p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t H = p_i \Delta p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t H = p_i \Delta p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t H = p_i \Delta p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t H = p_i \Delta p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t H = p_i \Delta p_i = -i\hbar \partial_i E = -i\hbar \partial_i$ MicroPython on LEON 20/21

 $-1/r_{J}d^{2}u/d\phi^{2} + u - GM/A^{2} - 3GMu^{2} = 0 \delta = 2GM/R \int \psi^{*}\psi dx = 1 P(x, t) = |\psi(x, t)|^{2} |\psi(x, t) = |\psi(x, t)|^{2}$ $\pm \mathbf{x} \times \mathbf{B} = -\hbar^2/2m\nabla^2\psi(\mathbf{x},t) \pm V(\mathbf{x})\psi(\mathbf{x},t) = E\psi(\mathbf{x},t) \ \mathbf{x}' = \gamma(\mathbf{x},t)$ $\partial_{\mu}J^{\mu} = 0 E_i = -1/c\partial A_i/\partial t - \partial_i \phi B_i = \epsilon_{ijk}\partial_j A_k F^{\mu\nu} = \partial^{\mu}A^{\nu} - \partial^{\nu}A^{\mu} \tilde{F}^{\mu\nu} = 1/2\epsilon^{\mu}$ $0 R^{\mu}_{\mu\rho\sigma} = \Gamma^{\mu}_{\mu\sigma,\rho} - \Gamma^{\mu}_{\mu\rho,\sigma} + \Gamma^{\alpha}_{\alpha\sigma} \Gamma^{\mu}_{\alpha\rho} - \Gamma^{\alpha}_{\alpha\rho} \Gamma^{\mu}_{\alpha\sigma} R_{\mu\nu} = R^{\rho}_{\mu\nu\rho} R = R^{\mu}_{\mu} G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$

 $\partial \mathbf{B}/\partial t \nabla \times \mathbf{B} = \mu \mathbf{J} + 1/c^2 \partial \mathbf{E}/\partial t \mathbf{F} = g (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2/2m\nabla^2 \psi$

 $^{\mu}_{\alpha \rho} - \Gamma^{\alpha}_{\nu \rho} \Gamma^{\mu}_{\alpha \sigma} R_{\mu \nu} = R^{\rho}_{\mu \nu \rho} R = R^{\mu}_{\mu} G_{\mu \nu} = R_{\mu \nu} - 1/2g_{\mu \nu} R = 0$ $\langle x \rangle = \langle \psi | x | \psi \rangle \Delta x \Delta y \ge \hbar/2 \ y_i = -i\hbar \partial_i \ E = i\hbar \partial/\partial t \ H = y^2/2m + V$

 $= J^{\nu} \partial_{\nu} \tilde{F}^{\mu\nu} = 0 ds^{2} = c^{2} dt^{2} - dx^{2} - dy^{2} - dz^{2} ds^{2} = g_{\mu\nu} dx^{\mu} dx^{\nu} g_{\mu\nu}$ $2GM/r)dt^2 - dr^2/(1 - 2GM/r) - r^2d\theta^2 - r^2\sin^2\theta d\phi^2 \Delta\nu \approx$

micropython.org

forum.micropython.org
github.com/micropython



 $r/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$ $= 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$ $1/c^2 \partial \mathbf{E} / \partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m \nabla^2 \psi$ $p^2 c^2 + m^2 c^4 \partial_\mu J^\mu = 0 E_i = -1/c \partial A_i$ $\beta_{\mu\nu\rho}^{\rho} R = R_{\mu}^{\mu} G_{\mu\nu} = R_{\mu\nu} - 1/2g_{\mu\nu}R = 0$ $/2 p_i = -i\hbar \partial_i E = i\hbar \partial/\partial t H = p^2/2m + V$ $-t \sinh \xi + x \cosh \xi \ t' = t \cosh \xi - x \sinh \xi \ ta$ $^{2}dt^{2} - dx^{2} - dy^{2} - dz^{2} dz^{2} = g_{\mu\nu}dx^{\mu}dx^{\nu}$ $1 - 2GM/r) - r^2 d\theta^2 - r^2 \sin^2 \theta d\phi^2 \Delta \nu \approx$ $r/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$ $\gamma T_0 u' = (u - v)/(1 - uv/c^2) p = \gamma mv$ $= 1 \Gamma_{\mu\nu\sigma} = 1/2(g_{\mu\nu,\sigma} + g_{\mu\sigma,\nu} - g_{\nu\sigma,\mu})$ $d^2 u/d\phi^2 + u - GM/A^2 - 3GMu^2 = 0 \delta =$ $1/c^2 \partial \mathbf{E} / \partial t \mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \hbar^2 / 2m\nabla^2 \psi$ $p^2 c^2 + m^2 c^4 \partial_\mu J^\mu = 0 E_i = -1/c \partial A_i$ $v^{\nu}v^{\sigma} = 0 R^{\mu}_{\nu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\alpha}_{\nu\sigma}$

 $r/r^3 \nabla \cdot \mathbf{E} = \rho/\epsilon \nabla \cdot \mathbf{B} = 0 \nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$

 $= \mu J + 1/c^2 \partial E / \partial t F = q (E + v \times B) - \hbar^2 / 2m \nabla^2 \psi$

$$\begin{split} &(1+2)M(r) = r^2M^2 - r^2M^2 - db^2 \Delta r = v_i GM(1/r_i - 1/r_f) - d^2u/d\theta^2 + u - GM/A^2 - 3GMa^2 = 0 + 2GM/R \int \phi^2 \psi dx = 1 P(x_i, 1) = \psi(x_i, 1)$$

D.P. George (George Robotics Ltd)

MicroPython on LEON 21/21