

## MICRO ROBOTS

## Microscopic robots that sense, think, act, and compute

Maya M. Lassiter<sup>1†</sup>, Jungho Lee<sup>2†</sup>, Kyle Skelil<sup>3</sup>, Li Xu<sup>2</sup>, Lucas Hanson<sup>4</sup>, William H. Reinhardt<sup>1</sup>, Dennis Sylvester<sup>2</sup>, Mark Yim<sup>3</sup>, David Blaauw<sup>2</sup>, Marc Z. Miskin<sup>1\*</sup>

Although miniaturization has been a goal in robotics for nearly 40 years, roboticists have struggled to access sub-millimeter dimensions without making sacrifices to onboard information processing because of the unique physics of the microscale. Consequently, microrobots often lack the key features that distinguish their macroscopic cousins from other machines, namely, on-robot systems for decision-making, sensing, feedback, and programmable computation. Here, we take up the challenge of building a robot comparable in size to a single-celled paramecium that can sense, think, and act using onboard systems for computation, sensing, memory, locomotion, and communication. Built massively in parallel with fully lithographic processing, these microrobots can execute digitally defined algorithms and autonomously change behavior in response to their surroundings. Combined, these results pave the way for general-purpose microrobots that can be programmed many times in a simple setup and can work together to carry out tasks without supervision in uncertain environments.

## INTRODUCTION

Natural microorganisms demonstrate the feasibility of building autonomous, intelligent systems at dimensions too small to see by eye, a fact that has fascinated roboticists for decades (1, 2). Ideally, a microscopic robot would retain all of the features that distinguish its macroscopic cousins from other machines: It would be able to sense, compute actions, support repeated programming, and manipulate or explore the surrounding world. Yet miniaturizing robots without forgoing many of these hallmarks has proven difficult.

At present, the smallest robots with fully integrated systems for sensing, programmable computation, and motion control sit at or above a millimeter (3), a size first reached more than 20 years ago (4). Further shrinking these early successes has been hampered by the fact that many of the physical laws that govern semiconductor circuits (5), energy storage (6), power transfer (7), and macroscale propulsion (8, 9) scale superlinearly with size, compounding into marked problems as robots attempt to shrink under submillimeter dimensions. As a result, fundamentally different approaches are required for truly microscopic robots (10).

Typically, roboticists circumvent the barriers to reaching the microscale by externally controlling locomotion through peripheral hardware, sacrificing programmability, sensing, and/or autonomy in the process (9, 11–13). These devices, although highly functional, remain tied to external equipment to make decisions and execute tasks, frequently cannot respond to their environment, and/or can only transition between a limited number of hard-coded behaviors on command. Even when microrobots do incorporate onboard electronics, achieving unidirectional communication and open-loop locomotion, they still cannot sense their surroundings and have yet to incorporate a real computer architecture that would support programmable decision-making (14). As a result, microrobots struggle with unknown environments and offer limited reconfigurability

after fabrication, blunting their usefulness in real-world applications (13).

Here, we take a different approach, directly addressing the challenge of building a robot that can sense its environment, compute digitally programmable actions, and change its behavior, all while fitting in a space that is too small to see with the naked eye. Our approach leverages semiconductor manufacturing to lithographically build the robot's body, actuators, and information systems massively in parallel. By optimizing the circuits, actuators, and fabrication protocols to match the physical constraints of working at small scales, we were able to shrink the volume of programmable, autonomous robots that sense, think, and act by 10,000-fold (3, 4, 15). At the same time, moving computation to the microrobot reduces both the cost and operational overhead to a bare minimum, paving a path to widespread adoption.

## RESULTS

## Challenges to scaling

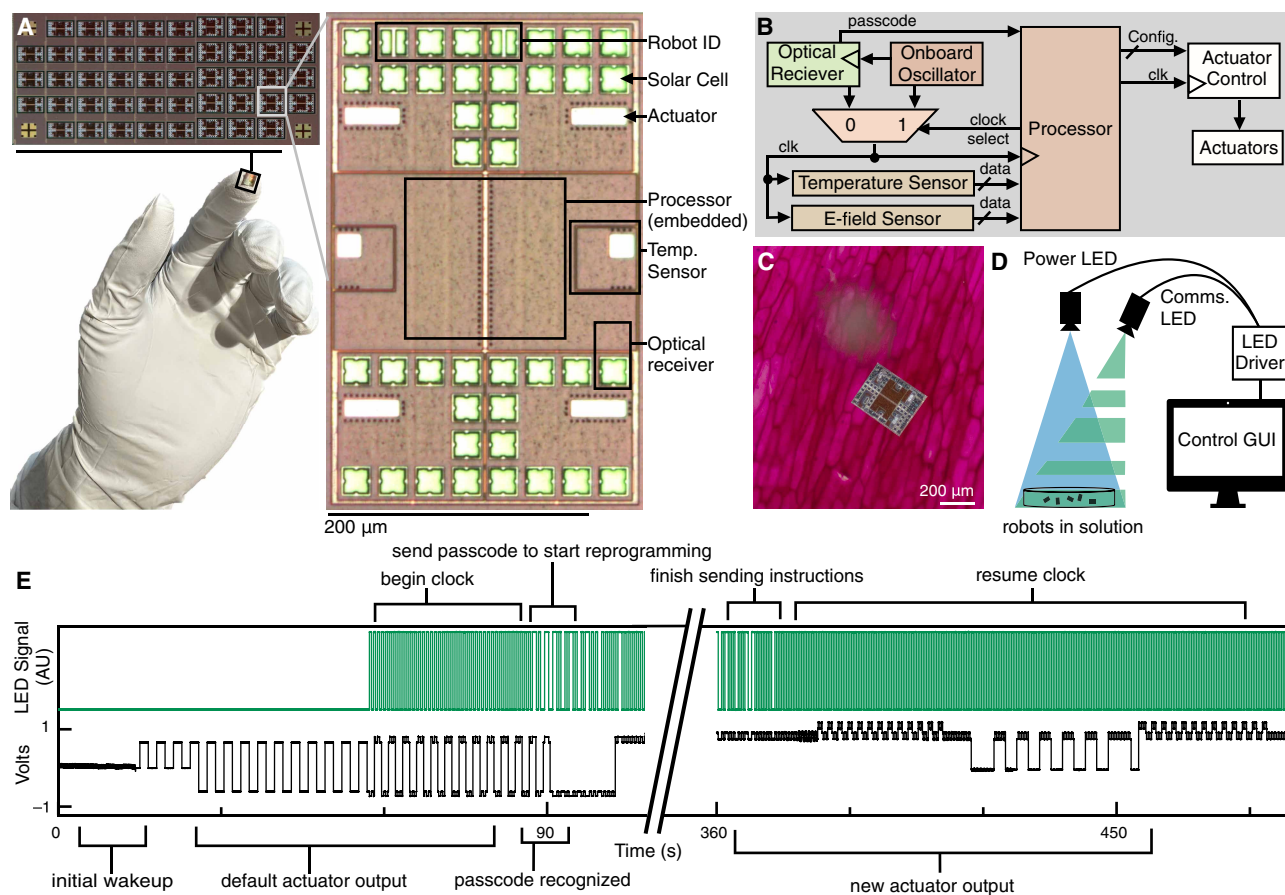
Although many of the essential components for a smart microrobot, including small-scale sensors, actuators, and information processing systems, have been demonstrated individually in ~100- $\mu\text{m}$  packages (16–20), integrating these systems into a single microrobot without violating constraints on power, size, manufacturability, or performance remains challenging. For instance, propulsion and computation are difficult to simultaneously accommodate with the limited power budget (~100 nW) of small robots. Likewise, the materials and processes used to build these systems must be compatible: Building the actuator cannot damage the electronics. Similarly, processes must be scalable to large numbers of freely released devices.

We addressed the challenge of integration with circuits and nanofabrication protocols tailored to autonomous microrobots. On the circuit side, we operated within a limited power budget of ~100 nW by building our robot in a 55-nm complementary metal-oxide semiconductor (CMOS) process and leveraging subthreshold digital logic. This process's high-threshold voltage reduced transistor leakage enough to support a variety of onboard circuits in a submillimeter package without violating power constraints. Specifically, Fig. 1 (A and B) shows that, in 210  $\mu\text{m}$  by 340  $\mu\text{m}$  by 50  $\mu\text{m}$ , we were able to fit photovoltaic (PV) cells for power, sensors for temperature, four

<sup>1</sup>Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. <sup>2</sup>Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. <sup>3</sup>Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, USA. <sup>4</sup>Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA, USA.

†These authors contributed equally to this work.

\*Corresponding author. Email: mmiskin@seas.upenn.edu



**Fig. 1. Overview of the microrobot circuits.** (A) A millimeter-scale chip of roughly 100 microrobots, resting on a gloved fingertip. Each microrobot contains several integrated pieces of microelectronics, spanning sensing, memory, processing, communication, and power (scale bar, 200  $\mu\text{m}$ ). These devices were fabricated together in a 55-nm CMOS process at a commercial foundry and were optimized for size and power. (B) A flowchart depicting the movement of data between the robot's subsystems. Incoming information from the sensors and communication system is used by the processor to update actuator outputs. (C) A composite photograph of a robot on top of plant cells, to show scale (scale bar, 200  $\mu\text{m}$ ). (D) A schematic illustrating how external LEDs can power and program the robot using light. To send instructions, an automated compiler modulated the optical power of a communication LED, with each flash encoding a bit in the program. (E) Experimental data showing the robot responding to an optical transmission. To prevent random light fluctuations from interfering with the robot's operation, the communication protocol featured a passcode sequence that a microrobot must see to accept an incoming program, store the data to its memory, and execute.

actuator control circuits, an optical receiver for downlink communication and programming, a processor, and memory. These subcircuits were optimized to minimize size and power consumption, were fabricated at a commercial foundry, and are detailed in our prior publication and/or Materials and Methods (19). We built two robot configurations that differed only in the number of PV cells, resulting in body widths of 210 and 270  $\mu\text{m}$ , although, as shown later, we found no notable differences between them in performance. At this scale, the robot's size and power budget are comparable to many unicellular microorganisms (21); Fig. 1C shows the robot beside plant cells for scale.

### Computation and communication

The fabrication processes used to achieve dense data storage in commercial electronics are incompatible with the low-leakage requirements of an autonomous microrobot. Consequently, we faced tradeoffs among area, memory, and power generation: Storing more bits increases leakage currents, which, in turn, requires more energy production. Table 1 shows the specific breakdown of area and power

for each of the robot's subcircuits. The processor consumed most of the power, requiring nearly a third of the robot's area to be devoted to energy harvesting to compensate. Even when using larger, low-leakage transistor-based memory, at this process node, we were limited to storing a few hundred bits.

To achieve digitally controlled behaviors despite this tiny memory size, we used a custom-designed complex instruction set computer architecture for the robot's processor, compressing useful robot actions into specialized instructions like "sense the environment" or "move for  $N$  cycles." This design is well suited for microrobots because it lifts memory demands from the processor without sacrificing functionality (see Table 2 for a full list of commands). Further, by adopting a fully functional computer, the programmable memory defines both the sequences of instructions and the robot's internal states. That is, unlike prior work on electronic microrobots (14), the way that states are updated can also be digitally reconfigured.

The processor's commands can be split into two groups: one for manipulating data and the other for implementing robot-specific

**Table 1. Custom-designed 11-bit instruction set for a microrobot on-board processor.** Rx, Ry, and Rz indicate arbitrary register file addresses; Imm, immediate value; Mx, arbitrary memory address; A, arbitrary instruction address; O, arbitrary address offset relative to the current program counter.

Instruction	Operation	Meaning
<i>Actuator</i>		
mot	Actuator motion ( $N$ )	Enable select actuators, then repeat programmed motion $N$ times
wav	$Mx \rightarrow$ Actuator motion	Modulate select actuator motion by Manchester-encoded pattern of data in $Mx$
<i>Sensing</i>		
ts	Temperature $\rightarrow$ $Mx$	Sense temperature, then store data in $Mx$
<i>Data transfer</i>		
mov	$Rx \rightarrow Ry$	Move data from $Rx$ to $Ry$
sb	$Rx \rightarrow Mx$	Store byte in $Rx$ to $Mx$
lb	$Mx \rightarrow Rx$	Load byte in $Mx$ to $Rx$
li	$Imm \rightarrow Rx$	Load $Imm$ to $Rx$
<i>Arithmetic</i>		
add	$Rx + Ry \rightarrow Rz$	Add $Rx$ and $Ry$ data, then store the result in $Rz$
addi	$Rx + Imm \rightarrow Rz$	Add $Rx$ data and $Imm$ , then store the result in $Rz$
sub	$Rx - Ry \rightarrow Rz$	Subtract $Ry$ data from $Rx$ data, then store the result in $Rz$
subi	$Rx - Imm \rightarrow Rz$	Subtract $Imm$ from $Rx$ data, then store the result in $Rz$
and	$Rx \& Ry \rightarrow Rz$	Bitwise AND for $Rx$ and $Ry$ data, then store the result in $Rz$
or	$Rx   Ry \rightarrow Rz$	Bitwise OR for $Rx$ and $Ry$ data, then store the result in $Rz$
sll	$Rx \ll Imm \rightarrow Rx$	Shift left logically for $Rx$ data of the amount of $Imm$ , then store the result in $Rx$
srl	$Rx \gg Imm \rightarrow Rx$	Shift right logically for $Rx$ data of the amount of $Imm$ , then store the result in $Rx$
<i>Unconditional jump</i>		
j	go to $A$	Jump to address $A$
bcnt	go to $-O$ ( $N$ )	Jump to address with offset $-O$ , then repeat the in-between instructions $N$ times
<i>Conditional branch</i>		
cmp	compare $Rx, Ry$	Compare $Rx$ and $Ry$ data, then set flag bits, eq/ne/gt/lt
cmpi	compare $Rx, Imm$	Compare $Rx$ data and $Imm$ , then set flag bits, eq/ne/gt/lt
beq	go to $+O$ if equal	Conditional branch on eq to address with offset $O$
bne	go to $+O$ if not equal	Conditional branch on ne to address with offset $O$
bgt	go to $+O$ if greater than	Conditional branch on gt to address with offset $O$
blt	go to $+O$ if less than	Conditional branch on lt to address with offset $O$

functions. To support the processor, we built a logic decoder for the 11-bit instruction set, a 32-entry 11-bit instruction memory, a register file with four 8-bit registers, and a 16-entry 8-bit data memory. For data manipulation, we included instructions for conventional arithmetic operations, such as addition (add), subtraction (sub), bitwise AND/OR (and/or), and shift left/right (sll/srl). We also added operations for control flow, including unconditional jumps and conditional branches based on comparisons between two registers. Last, we included a move (mov) command to transfer data between registers and store (sb) and load (lb) commands to move data between registers and memory.

The remaining instructions interfaced onboard data and the physical world by controlling the sensors and actuators. Namely, we included a command for motion control (mot) that can drive a user-specified voltage sequence to the actuators; a sensing instruction (ts) that can measure the temperature around the robot and stores the value in memory; and a communication instruction (wav) that can take the value in a register, Manchester-encode it, and then modulate selected actuators to output the data. Each of the robotic instructions required multiple clock cycles to complete. However, to

simplify programming and reduce program size, we designed these commands to halt instruction execution so that, from the programmer's point of view, they appear to complete in a single cycle. In effect, this design compresses into a single command what would, otherwise, require dozens of instructions, making it possible to perform useful tasks with minimal memory.

To further reduce the required memory size, we implemented progressive, multistep programming via the communication downlink, using the base station as a buffer. First, we loaded the microrobot with an initialization program to configure the desired actuator states. Second, we transmitted task programs, defining the robot's operation.

We sent both the initialization and task programs to robots using an optical communication link, shown in Fig. 1D. Two light-emitting diodes (LEDs) of different wavelengths illuminated the microrobots: one for powering each robot's onboard PVs and one for communication. The communication LED sent data by first transmitting an initial sequence of illumination flashes, called a passcode. This sequence instructed the robot to receive subsequent flashes as bits and write them to its instruction memory bank (Fig. 1E). We

**Table 2. Estimated power and area usage.** Most of the robot's power is dominated by the onboard computing system in the form of memory leakage. Conversely, most of the area is used for energy harvesting with PV cells. n/a values are denoted for circuit elements that either generate (PVs) or do not consume power (empty space). Dashes indicate that specific subdivision calculations were not performed.

Subpart	Power (nW)	Power (%)	Area ( $\mu\text{m}^2$ )	Area (%)
<i>Processor</i>				
Total	15.17	93.2%	19,252	26.0%
Memory	–	–	7689	10.4%
Logic	–	–	11,563	15.6%
<i>PV + leg</i>				
Total	n/a	n/a	39,808	53.8%
Harvesting PV			24,880	33.6%
Disabled PV units			9952	13.4%
Leg pads			4976	6.7%
<i>Power-on reset</i>				
Total	0.00	0.0%	181	0.2%
<i>Optical receiver</i>				
Total	0.18	1.1%	1401	1.9%
<i>Temperature sensor</i>				
Total	0.19	1.2%	674	0.9%
<i>Electric sensor</i>				
Total	0.22	1.3%	674	0.9%
<i>Clock generator</i>				
Total	0.51	3.2%	185	0.2%
<i>Supply decap</i>				
Total	n/a	n/a	286	0.4%
<i>Singulation guard and empty space</i>				
Total	n/a	n/a	11,546	15.6%
Chip total	16.28	100.0%	74,006	100.0%

added the passcode to prevent random fluctuations from altering the robot's state, and each robot was designed to recognize two: a global passcode, common to all robots, and a type-specific passcode that enables addressing specific subsets of robots.

We fully automated the process for generating optical instructions using a graphical user interface (GUI) and a custom-built printed circuit board, enabling any individual, regardless of experience level, to program the robot. Yet once instructions were written to the memory, the robot's behavior was fully autonomous: The robot computed its actions on the basis of the onboard program and its sensor data (Fig. 1B) without further user input.

## Propulsion

Although there are many mechanisms for motion at the microscale (20, 22), compatibility with onboard electronics requires low-current (<100 nA) and low-voltage (~0.1 to 1 V) operation. To our knowledge, two actuators fit these constraints: surface electrochemical actuators (20, 23, 24) and electrokinetic propulsion (25). The latter is the subject of this paper, whereas the former is a topic of ongoing work.

The governing mechanism for electrokinetic propulsion has been discussed in detail in our prior work (25); we review it briefly here. As shown schematically in fig. S1, the robot, which must be immersed in a fluid, passes a current between oppositely biased electrodes. Mobile ions surrounding the robot and nearby surfaces

move in response to this field, dragging the fluid with them. This establishes a flow, which, in turn, causes the robot to move at a speed proportional to the applied electric field. By manipulating the electric field through spatial patterns of active electrodes, the robot can travel in different directions and/or turn with a power expenditure of ~60 nW at this size.

Compared with alternatives, electrokinetic propulsion offers several advantages. First, it is straightforward to implement: The electrodes themselves are simple layers of platinum, requiring no moving parts or complex fabrication steps. Consequently, these actuators are robust, lasting upward of several months, and fabrication can be done massively in parallel using fully lithographic patterning (25). Second, the electrical signal is a direct current (dc) voltage, requiring no upconversion or complex temporal sequencing (as would be needed for legs). This reduces the circuit overhead needed to control the robot's behavior to a bare minimum, leaving more space for sensing, memory, and computation.

To integrate these actuators and release the robot from the underlying silicon wafer, we carried out a series of lithographic, low-temperature, back-end fabrication steps, as shown in Fig. 2. This protocol improved on prior work for microrobot release based on specialized silicon-on-insulator wafers (14, 23) by generalizing to wafers built at arbitrary semiconductor foundries in a wide range of process nodes. Briefly, we added an oxide border around each microrobot

body on the layout sent to the foundry to avoid the presence of metal structures typically placed for planarity concerns in commercial semiconductor processes. After full assembly of the circuits, we thinned the backside of each chip to 50  $\mu\text{m}$  using a combination of mechanical and plasma etching. Then, we deposited and patterned chrome layers on the top and bottom of the chip to act as masking layers and etch barriers, respectively. From the top, we etched through the oxide border to the underlying silicon with an inductively coupled plasma followed by etching through the remaining silicon wafer with deep reactive ion etching until we hit the bottom layer of chrome. This stopping layer both arrested the etch and supported the released robots. Last, we wet-etched the residual chrome, releasing the robots into solution. Typical yields exceeded 50%.

Released chips with embedded actuators formed microscopic robots, each with tightly integrated systems to sense, compute, communicate, and move. As a first test, we characterized robot locomotion with simple programs that directly set actuator polarity and thus propulsion direction. Under digital control, a four-electrode robot has 14 unique configurations that generate motion (2 of the 16 polarity states are degenerate, corresponding to all electrodes at the same voltage). As shown in Fig. 3, the 14 states group into four behaviors: translations along the major or minor axis, rotations, and arcs. Each of these groups is determined by the number and arrangement of positive polarity electrodes on the robot, with robots differing only by rotational symmetry within the same group. Statistics collected over 56

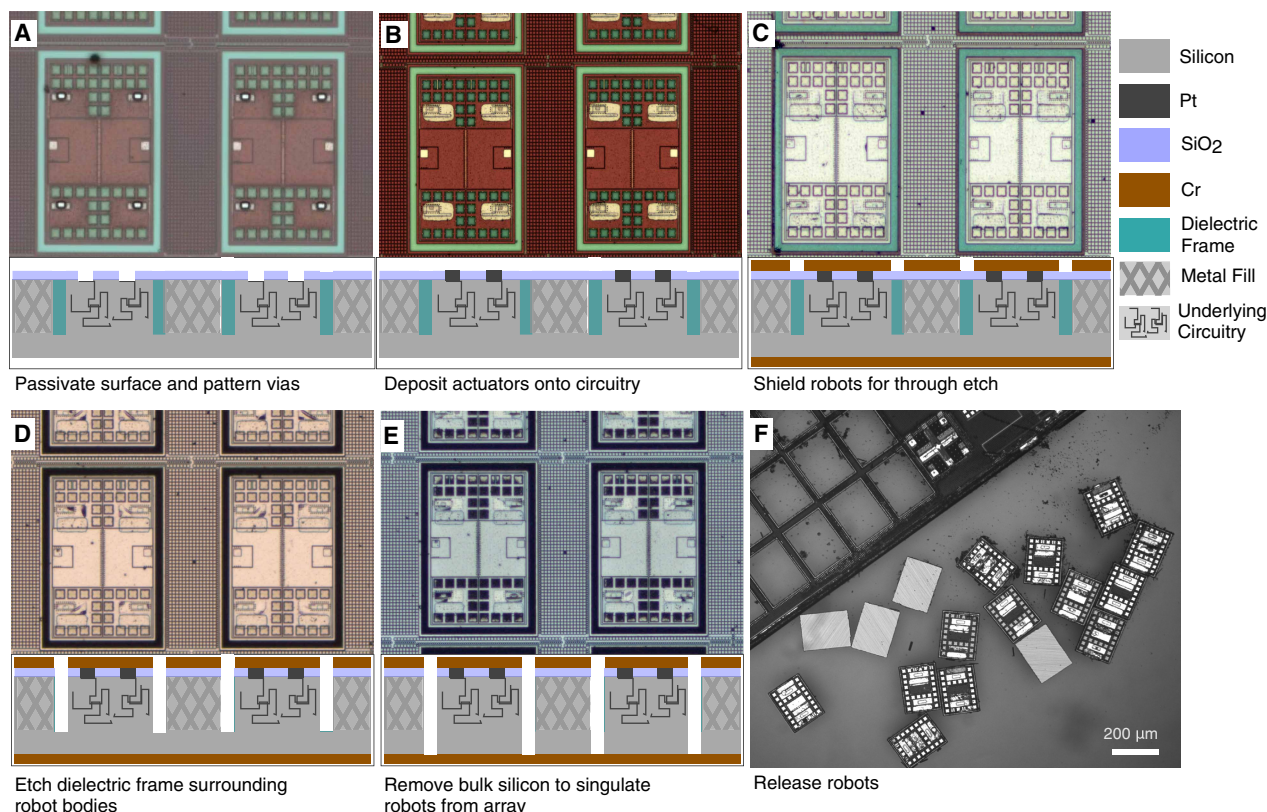
experimental trials show that typical speeds range from 3 to 5  $\mu\text{m/s}$  for translations and 0.1 to 0.3 $^\circ/\text{s}$  for turns, with a statistical variability that depends on imposed behavior (Fig 3B). Further discussions on the role of solution conductivity, polarity, current, and expected improvements in speed via advanced circuitry are in Materials and Methods.

Arcs, turns, and translations offer full control over the robot's three degrees of freedom and thus can be strung together to trace out arbitrary, user-specified paths in a plane. Movies S1 to S3 show a base station transmitting sequences of actuator commands to a robot and the robot updating its motion as it receives each message. Specific paths can also be given to specific robots by prefacing different passcodes to the instructions: Figure S2 and movie S4 show two robots executing different sequences of motions, each responding only to its own, type-specific set of instructions.

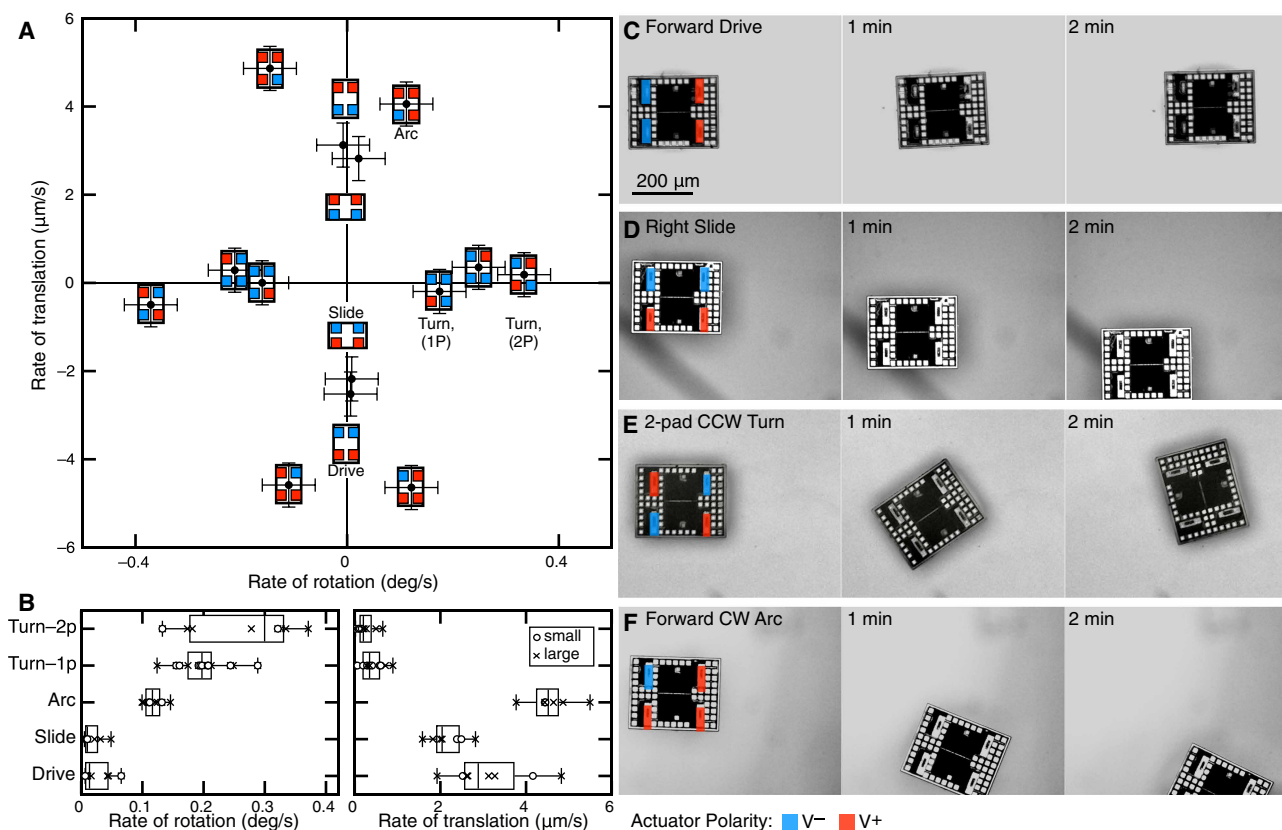
### Sensing, feedback, and control

With an onboard computer, motion primitives can be turned directly into behaviors; on-robot programs can transition between locomotion modes on the basis of sensor data. Here, we explored two different closed-loop behaviors with robots adapting to changes in temperature. In one case, the robot was tasked with responding to a changing temporal pattern; in the other, it was tasked with climbing spatial gradients.

For the temporal task, we exposed the microrobots to a heating environment and programmed them to report the current temperature



**Fig. 2. Integrated fabrication and release.** We transformed foundry-fabricated wafers into robots using the following steps. (A) We passivated the surface using oxide layers and patterned metal interconnects to the underlying electronics. (B) The actuators were deposited and then wired to the electronics through the interconnects. The remaining steps released the robots (C), first shielding them with a metal hard mask, then (D) etching through the oxide layers of the wafer, and (E) lastly removing the underlying silicon. (F) When finished, the hard mask layer was removed, releasing robots into solution en masse. Typical yields were roughly 50%. Scale bar, 200  $\mu\text{m}$ .



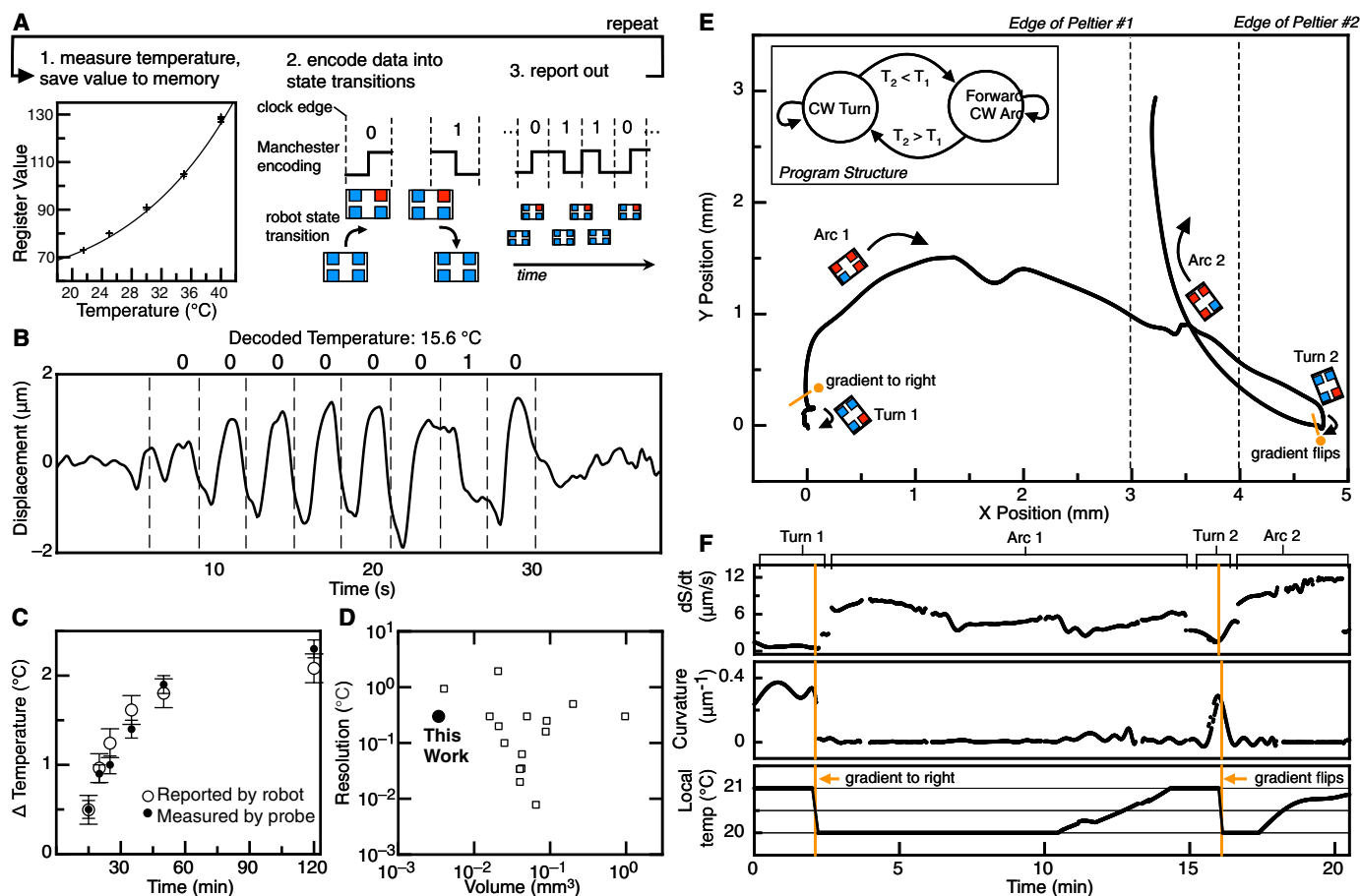
**Fig. 3. Reprogrammable locomotion.** (A) The robot's kinematic degrees of freedom, namely, its signed rate of rotation and forward velocity, depended on the electric field produced in solution by passing current through its electrodes. By setting different electrode polarities (red, positive and blue, negative), we generated four classes of robot motion: translations along the major and minor axes, turns, and translation with rotation, here called arcs. Turns could be achieved by either holding one electrode high (denoted as 1P) or holding two electrodes high along a diagonal (denoted as 2P). The robot's motion generally respects symmetry, with states differing only by rotations or reflections yielding similar velocities and rotation speeds. Data points correspond to individual experiments (14 total), with error bars quantifying systematic uncertainty from microscopy drift and ambient fluid flows. (B) Detailed statistics of robot motion show the device-to-device variation over 56 independent experiments. Boxes span the interquartile range, with a vertical line at the median; horizontal bars span the minimum-maximum range. Data collected for the two robot sizes (large and small) show no notable difference. (C to F) Montages of the four representative body motions over 2 min. Scale bar, 200  $\mu\text{m}$ .

by modulating their motion. Specifically, each microrobot was tasked to measure temperature with its onboard sensors, digitize the value, and transmit the result back to a base station by switching the polarity of the front-right actuators in a Manchester-encoded sequence (Fig. 4A). We then decoded the resulting body movement to read out the robot's temperature measurement in bits (Fig. 4B). Because of our custom instruction set, most of this program was implemented with just two commands: "ts" handled the sensing, and "wav" encoded and transmitted the data. The former executed over 30 to 500 ms; the latter ran over roughly 10 to 100 s (Fig. 4B), with the exact durations adjustable by the user. We validated the full program by placing robots in a bath of cooled solution that passively warmed to room temperature. When we compared the robot's reported measurements to simultaneous thermocouple readings, we found agreement, as shown in Fig. 4C.

As a sensor, the microrobots demonstrated high performance (16, 26–39). Figure 4D plots sensor resolution (here, set by quantization) against volume for a variety of state-of-the-art digital thermometers. Our robot Pareto dominates the field, offering a 0.3°C resolution in  $<1 \text{ mm}^3$ , a consequence of our release strategy that accommodates thinned, submillimeter devices. The handful of sensors that have achieved better resolution occupy at least an order of

magnitude more volume (see table S1 for additional metrics). As a result, our sensor could probe thermal gradients at smaller spatial scales or in more restrictive geometries, like microfluidic chambers or capillary tubes.

To show adaptation in response to spatially varying environments, we programmed microrobots to climb thermal gradients. Using thermoelectric heat pumps, we introduced a temperature gradient in the solution bath and programmed the robot to execute an arcing motion to search for warmer regions if the current temperature reading was lower than the previous one. Alternatively, if the temperature reading was warmer, the robot was programmed to switch to a turning state, pivoting without translation. We chose these two states because they presented a distinct signal to validate the behavior and enabled the robot to broadly explore space. Figure 4D and movies S5 and S6 show the experimental results. Initially, there was no imposed gradient, and the robot correctly turned in place. When the heat pump was turned on to cool the local area, the robot switched to an arcing state and explored the workspace until finding a warmer region, where it resumed rotating. To further validate the program, we reversed the direction of the temperature gradient. In response, the robot transitioned back to arcing, traveling in the



**Fig. 4. Motion controlled by sensor feedback.** (A) A program to sense temperature and transmit the result using Manchester-encoded motion signals. The robot sampled temperature and encoded an 8-bit register value for each reading. Plotted cross symbols represent individual measurements without error bars from a single robot at known temperatures, which were used to calibrate robot-to-robot sensor variation. Total of 42 total measurements, ~5 per temperature (see corresponding dataset). (B) Robots sent data by moving their center of mass, which we tracked and decoded. Transmissions contained 8 bits sent at a known frequency, enabling segmentation into time windows (dashed lines), each containing a transition. Bits were determined on the basis of whether the signal led with a high or low state and then converted to decimal. (C) By placing robots in a warming bath, we could compare their measurements (open circles) to those made with a probe (filled circles), finding agreement. Error bars reflect the resolution of the robot (quantization-limited, ~0.3°C, full width) and probe; deviation from ground truth was ~0.2°C. (D) Temperature resolution versus total volume for our results and existing works (16, 26–38) (details in table S1). (E) A program for climbing temperature gradients. The robot either explored the environment by executing an arcing motion when the temperature dropped below a previously recorded value (inset) or turned in place to hold its position. When a thermal gradient was applied, robots arced until reaching a warmer region, then switched to turning. Reversing the gradient caused robots to resume arcing. (F) Curvature, speed, and estimated local robot temperature from thermal imaging confirmed that the transitions took place in response to new, local temperature measurements.

opposite direction. At times, the robot moved faster in this experiment than in the static tests of Fig. 3. We hypothesize that this was a consequence of thermophoretic effects, like enhanced mobility; additional thermoelectric fields; or transient behaviors due to controlled switching between two states, although we leave a thorough investigation for subsequent work (see Supplementary Methods for further discussion).

The combined capacity to explore space and report temperature at high precision makes microrobots potentially suited for measurements in life sciences (40): Robots can fit in submillimeter spaces; record with sub-1°C accuracy; report over the full range of temperatures in mesophile cell biology (Fig. 4A); and reposition over time as cells grow, die, or restructure. Further, the robots shown here could interface with biological systems by placing their aqueous environment above a target substrate and allowing thermal equilibration. By allowing heat to flow between the environments, accurate readings

can be obtained without direct interaction, thereby bypassing biocompatibility issues. Future work could also explore imaging methods like dynamic light scattering or photoacoustic tomography, using digital encoding to eliminate microscopes, improve resolution, or widen the sensing area.

## DISCUSSION

Many other microrobots have achieved taxis-like behaviors (41), often using vastly simpler analog effects. Such devices represent an elegant counterpoint to our approach. Rather than fabricating sophisticated circuits, one can achieve similar goals with appreciably lower manufacturing and design complexity by embedding responsiveness in the material itself (12).

However, full, programmable autonomy with local sensor feedback is a unique capability for microscopic robots that brings distinctive

benefits. First, many current microrobot platforms require large pieces of specialized laboratory equipment, like magnetic coils (42), ultrasound transducer arrays (43, 44), or cell culture environments (45). This overhead restricts use to experts with bespoke, expensive equipment, even if the cost per robot is low. By contrast, we require only a controllable light source for power and programming because information processing takes place on the robot. The result is high-level functionality using only commonplace parts. Second, current microrobots that can sense or control their behavior execute fixed, design-specific tasks (12–14). In contrast, our robot integrates a functional digital computer in a few hundred micrometers of space. As shown, digital programming and onboard computing allow a single, general-purpose microrobot to carry out a range of tasks that can be reconfigured on demand, after fabrication.

This flexibility is critical as microrobots move toward applications where the environment, task, or robot state is unknown or changes. For example, in the future, more advanced electronically integrated microrobots could aid in targeted drug delivery, releasing drugs in response to local sensor cues like biochemical markers or temperature changes rather than global commands. In telemetry, onboard computation could digitally encode data, like the Manchester encoding shown here, to make communication more robust to noise. In nanomanufacturing, programmable microrobots could use passcode-based communication to receive, monitor, and update instructions as they work.

Although the rudimentary robots demonstrated here would need further advances like new actuators (20) or power transfer schemes (6, 7) to realize these long-term goals, our circuits and computer architectures provide a foundation poised for growth. For example, onboard memory could be increased ~100-fold by adopting a more advanced process node, provided that the threshold voltage remains high enough to suppress leakage. This expansion would facilitate more complex programs, approaching thousands of lines of code, and more sophisticated autonomous behaviors. Building on the work here, programs can already be made unique to individual robots via integrated passcodes (movie S4). Even without direct inter-robot communication, this allows a central controller to assign distinct tasks or emulate multiagent coordination. By driving and measuring currents in solution, circuits could facilitate spatial localization or fluid-mediated communication. Last, subcircuits optimized for propulsion could improve robot speed 10-fold (see Materials and Methods) or, alternatively, drive other electronically integrated actuators suited to demanding environments like *in vivo* operation (20, 23).

If pursued, these benefits could be achieved without incurring lab overhead or markedly increasing the cost per machine, which, scaled to production, falls at roughly a penny per robot (see Materials and Methods). The combination of versatility, ease of use, and low cost offered by onboard information processing could find uses in fields ranging from scientific studies on the physics of living systems (46, 47) to nanomanufacturing (48), microsurgery, and drug delivery (22, 49).

## MATERIALS AND METHODS

### Microrobot fabrication

Microrobot fabrication can broadly be grouped into four main steps. For brevity, we summarize them here and provide complete details, with tools, processing steps, characterization, materials, and vendor information, in the Supplementary Methods.

The steps for robot production were as follows. First, we encapsulated the electronics in a protective layer of oxide and put metalized holes through the oxide to enable connections. Second, we added platinum metal layers that the robot can use for electrokinetic propulsion. Third, we etched through an oxide border around the robots and the underlying silicon wafer to a supporting metal layer. Last, we dissolved a series of supporting metal layers to release robots into solution *en masse*.

### Using robots after release

In our prior work, we found that electrokinetic propulsion could be achieved in a variety of different solution environments, including deionized water, weak acid/base solutions, and weak hydrogen peroxide, and on a range of substrates, such as glass, metal, and polymers (25). The primary role of the solution environment is to set the proportionality factor between the applied field and robot speed or the effective mobility. Although this effect is minor (the highest mobility that we have identified, thus far, differs from the lowest by only fourfold), we found that working in 5 mM hydrogen peroxide provided one of the largest mobilities and tolerates variance in solution conductivity from gas absorption or other unintentional solution contamination.

Accordingly, we used 5 mM hydrogen peroxide for all experiments in this study, prepared by micropipetting 100 ml of deionized water and 25  $\mu$ l of 30% hydrogen peroxide (Fisher Chemical H325-500) into clean 100-ml glass graduated cylinders. After mixing, the solution was poured into sterile polystyrene petri dishes (Greiner Bio-One 632181), robots were pipetted into the dish, and the dish was loaded onto a microscope for imaging.

During experiments, we monitored solution conductivity, because unintended increases can reduce electric field strength and thereby slow propulsion. Using an ultrapure water meter (Hanna Instruments HI98197 EC/Resistivity with probe HI763123), we found that conductivity gradually increased over time, which we attributed to dissolved atmospheric carbon dioxide. However, this process typically saturated around 450  $\mu$ S/cm, which was low enough to support effective microrobot motion.

### Characterization of PVs

We measured the on-robot PVs' current-voltage characteristics to extract the open circuit potential, short circuit current, and efficiency. To electrically access the PV, we reprogrammed the robot to a dc forward drive state, directly connecting the actuator electrodes to the positive and negative terminals of the onboard PV (fig. S3). We then connected these terminals to a source meter (Keithley 2450) and measured the current required to hold the robot at a given bias. We found a short circuit current of roughly 1  $\mu$ W, an open circuit potential of roughly 0.8 V, a responsivity of 0.34 A/W, and a conversion efficiency of about 10%, all of which are typical for PVs made in a CMOS process.

### Improving locomotion via circuits for electrokinetic propulsion

The order of magnitude for electrokinetic propulsion speed is set by the applied electric field in the solution. In other words, the robot's speed scales in proportion to the applied current output by the robot's electrodes and is inversely proportional to the solution conductivity, as shown in our prior work (25). Thus, by increasing current supplied at the electrodes, we could markedly increase propulsion

speed. To do so, subsequent designs could operate closer to the voltage at which hydrolysis takes place ( $\sim 2$ -V electrode to electrode). In this regime, current scales exponentially with further increases in voltage, presenting a “short circuit” to the power supply and allowing the robot to directly control its speed by simply modulating the current limits.

In prior work, we found that robots operating near the water window could move at 1 mm/s when applying currents on the order of 10 to 100 nA. This current scale is feasible, given that the robots shown here already expend  $\sim 60$  nA for actuation. The primary limitation of the current circuits is that the operating voltage is too low ( $\sim 1$  V), presenting a high solution impedance to each electrode.

Last, we note that robot locomotion in our current design does not respect parity inversion of the applied voltage. That is, robots display different behaviors depending on the number of positively biased electrodes. We attribute this result to different faradaic reaction rates on the platinum electrodes for forward and reverse bias configurations. Specifically, we measured an asymmetry between the forward and reverse solution impedance during cyclic voltammetry, as seen in fig. S4 (sweep rate of 10 mV/s). Because it is the field within the solution, not across the electrode, that drives propulsion, this asymmetry can lead to different propulsion states with more current flowing (and thus faster fluid flows) in states that wire more electrodes in the positive direction. This effect could also be mitigated by increasing the operating voltage to enable direct control of the electrochemical current.

### Illumination conditions for operating robots

The optical receiver on board, based on silicon PVs, is sensitive to a range of different wavelengths, spanning roughly 375 to 1200 nm. To simplify data analysis, we used two different wavelengths for power and communication and filtered out the modulating communication wavelength when imaging. This enabled us to monitor the robot without having the background intensity fluctuate when we sent data. This work used 475 nm for static power illumination and 565 nm for communication, although these specific wavelengths were arbitrary. It should also be noted that the receiver is largely agnostic to wavelength and thus could be powered and programmed using a single LED if desired.

We tested reprogramming at various levels of illumination for both power and communication. Both channels were produced using a multiwavelength LED light source (Thorlabs CHROLIS), allowing us to control intensity with an analog voltage signal connected to a breakout box (Thorlabs BBC1). We measured illumination power incident on the workspace with a Thorlabs PM100D optical power meter (S120C 400- to 1100-nm 50-mW meter) fitted with a 10- $\mu\text{m}$  pinhole (Thorlabs P10K) to fix the detector area to a known value. We found that the static light signal intensity used for power needed to exceed  $200 \text{ W/m}^2$  to support onboard processing and locomotion. Similarly, the robots tolerated a maximum intensity of  $3000 \text{ W/m}^2$ , beyond which unwanted electron-hole pair generation disrupted circuit operation. Figure S5 shows other viable power and communication intensities, indicating successful reprogramming with a filled square. For this work, we operated in the middle region of the plot, using  $600 \text{ W/m}^2$  for power,  $1000 \text{ W/m}^2$  for communication, and a combined peak incident power of  $1600 \text{ W/m}^2$ . Of note, the range of power fluxes required for operation, spanning 200 to  $2600 \text{ W/m}^2$ , overlaps well with one standard “sun” of optical power ( $1000 \text{ W/m}^2$ ).

We used two different experimental setups for imaging the robots: one based on an upright microscope and the other based on macrophotography with a digital single lens reflex camera. The former facilitated high magnification; the latter offered a large field of view. For the microscope setup, we coupled the LED light source into an Olympus BX51-WI microscope using a liquid light guide (Thorlabs LLG-03-4H) and collimator (Thorlabs SLSLLG3). The microscope was equipped with a 4 $\times$  Olympus objective (PLN Plan Achromat Objective 1-U2B222) and used a 70:30 beam splitter (Thorlabs BST10R) to produce reflected light micrographs. To acquire images, we used a Scientifica SciCam Pro camera controlled by the MicroManager software package. Annotated images of the upright microscope setup are available as fig. S6.

The macrophotography setup used the same LED light source in a free-space configuration. We positioned a Canon EOS 5D Mark IV camera with a macro lens (RF-Mount 100 mm f/2.8 L Macro IS USM Lens) over a workspace using optical posts (Thorlabs PX series) and an adjustable macro focusing rail (Oben FRM-7 L). The LED output was projected through a collimator onto the workspace at a slight angle. An annotated image of this setup is available as fig. S7.

### Detailed description of locomotion data collection

We used our two imaging setups to obtain experimental movies of robots in solution. For the data in Fig. 3, we recorded robots over half-hour intervals at 10 frames per second. For the data in Fig. 4, we recorded roughly one frame every 10 s, using a reduced frame rate to accommodate the large image size. We obtained datasets for the robot’s speed, position, and orientation by binary image processing on the micrograph sequences. Given the high contrast between the robot and the background, we manually defined a threshold intensity on each movie for classifying pixels and used ImageJ to find the center of mass and orientation through ellipsoid fitting. To extract rates, we fitted these data to either lines (as in Fig. 3) or splines (as in Fig. 4, where the velocity is dynamic) and took analytical derivatives of the fit model.

### Detailed power and area budget

We optimized the robot’s circuitry for area, such as selecting a sub-threshold temperature sensor (50) and a leakage-based optical receiver (51). In general, to address power, it is necessary to use older technologies that reduce leakage, which creates a tradeoff with intelligence. As noted in Results, most of the robot’s area was devoted to energy harvesting ( $\sim 33\%$ ). The next largest contribution was the processor, which occupied roughly 25% of the body and consumed almost 90% of the robot’s power. The remaining sensors, clocking, and housekeeping circuits together took up less than 10% of the total power and area. These unequal proportions were consequences of operating at slow clock rates, a regime where most power loss is due to leakage currents in transistor memory.

In general, more memory requires both adding circuits to store the data and adding PV cells to support higher power demands. For the specific 55-nm process used here, leakage currents required a roughly 1:1 area allocation to meet the power demands at the targeted illumination intensities of roughly  $1000 \text{ W/m}^2$ . In other words, doubling the amount of memory on a robot would require an additional  $8000 \mu\text{m}^2$  for the memory cells plus an extra  $8000 \mu\text{m}^2$  of space for added power harvesting. Note that this ratio is sensitive to CMOS process technology. Future designs should thus compare the

actual area of a bit, after accounting for leakage effects, instead of the mere size of a circuit component, when identifying a useful process node.

### Detailed description of the communication protocol

Immediately after illumination, the robots were designed to execute a hard-coded configuration setting program, commonly called a bootloader. Once this program terminated, the robot demonstrated a default behavior, a sequence of 32 oscillations between front and rear electrodes followed by an ~10-s pause period before repeating, that we used to determine whether a given robot was viable before carrying out experiments.

We overwrote the default state using the optical link. For ease of experimentation, we used a Python GUI to compile desired behaviors into the corresponding binary bit stream and fed them at a user-set frequency of 0.5 Hz into the analog control port for the communication LED. A Raspberry Pi Pico handled the digital-to-analog conversion. When programming, the robot's onboard optical receiver was designed to phase-lock to the modulated LED signal after detecting the correct passcode sequence. Afterward, the optical receiver actively monitored the incident light power, set to begin rewriting memory upon detecting a passcode, as mentioned in Results.

After data transmission, we set the communication LED to oscillate at a fixed clock frequency. This allowed the robot to either run instructions using its onboard clock or tie its processor to the global oscillating signal to improve timekeeping accuracy. Switching between the global or local clock was specified in the program instructions and dynamic. That is, the robot could execute some instructions using the local clock and then switch to the global clock for others.

### Setting up and measuring thermal gradients

We produced thermal gradients for the temperature gradient climbing experiments using Peltier heat pumps (Digikay 926-1354-ND) driven by a Keithley 2450 source meter. We chilled the environment with the pump underneath the solution to prevent convective flows that could otherwise complicate robot motion. To calibrate the conversion factor, we independently measured the solution temperature at different voltage set points for the pump using a K-type thermocouple (Fluke 116 True RMS Multimeter). To measure temperature over a large area, we used a thermal imaging camera (FLIR ETS320) to estimate spatial gradients. Because this tool has high sensitivity but poor accuracy, we augmented the imager data with measurements from a thermocouple. That is, to produce the estimated gradients in Fig. 4, we only used gradient information from the imager and calibrated to the thermocouple measurements at specific sites.

### Probing microscopic robots

The robots shown here were only able to supply 100 nA at roughly 1 V from their PV cells, requiring a typical input impedance to be much greater than 10 megohms when probing. We used a Burr-Brown INA116 ultralow input bias current instrumentation amplifier with a 3-fA input bias current to read potentials from released robots in solution or in air. The amplifier was connected to robots through insulated microprobes (Microprobes for Life Science PI20035.0A3) that were held and translated using Scientifica PatchStar micromanipulators. Data from the amplifier was recorded directly with a USB oscilloscope (Picoscope 5442D). A circuit schematic, with  $V_{in+}$  and  $V_{in-}$  denoting the robot electrodes, is shown in fig. S8.

### Decoding robot data transmissions

To interpret digital signals sent by the wav command (movie S7), we applied a standard Manchester decoding procedure to the data extracted from the robot's center-of-mass motion, as shown in Fig. 4B. We high-pass-filtered signals before analysis (single-pole filtering with a corner frequency of ~10 s) to isolate the fast-switching motion associated with communication. Each transmission consisted of 8 bits sent at a known clock frequency. This allowed us to segment the resulting signal into eight uniform time windows, one per bit, such that a transition must occur within each window, as depicted by the dashed lines in Fig. 4B. To decode each bit, we checked whether the high state preceded or followed the low state within each window, corresponding to a transmitted 1 or 0, respectively. We then converted the final bit string into decimal, giving the value in the robot's register.

### Calibrating the temperature sensor and quantifying accuracy

To characterize the robot's temperature sensor, we placed robots on a heat pump alongside a thermocouple, imposed a temperature condition, and repeatedly measured what the robot reported. This protocol yielded calibration curves (as in Fig. 4A) and statistical distributions of robot measurements at each temperature and could be done on chips before release or using individual, freed robots. Of note, we did not fit the register values to a linear trend; we used a circuit-derived model (50). Briefly, the register value compares the frequency of a temperature sensitive oscillator with a reference oscillator. The former depends exponentially on environmental temperature, giving the register value,  $R$ , the functional form

$$R = \exp[a_1 + a_2/T]/f_c \quad (1)$$

where  $f_c$  is the reference clock frequency,  $a_1$  and  $a_2$  are calibration parameters, and  $T$  is the temperature in Kelvin. We fit a polynomial by transforming the raw data with the equation

$$R' = T \times \log[Rf_c] = a_1T + a_2 \quad (2)$$

This can be done with either a linear model or, to account for small drifts in the reference oscillator, including a quadratic correction that is held constant across devices from the same fabrication run. That is, the quadratic correction expresses the register as

$$R' = a_1T + a_2 + a_3T^2 \quad (3)$$

with  $a_3$  fixed across devices. Using Eq. 3 not only is more accurate but also requires a one-time bulk calibration from a small set of samples from the same batch. In either case, only two measurements are required per individual device.

To compute the inaccuracy, we subtracted the ground truth from the robot's temperature measurement; distributions are shown in fig. S9. Accounting for the significant digits from the register, we found that the sensor deviates by ~0.4°C (0.2°C) using the linear (batch-calibrated quadratic) model. In general, the error was dominated by quantization: We typically found just two reported register values at a given temperature.

### Cost of a robot

Robots in this work were purchased from Fujitsu as part of a multi-project wafer run at a price of ~10,000 USD/mm<sup>2</sup>. This resulted in roughly 100 chips, each of which contained roughly 100 robots. Thus, for the work here, we paid on the order of 10 USD per robot.

We note that, although this service no longer exists (the Fujitsu lab has since been acquired), a comparable service is offered by Muse Semiconductor, which lists current pricing on the internet, allowing up-to-date cost estimates. If scaled up to production, the cost would decrease as subsequent runs see a discounted price because of previous investment in the photolithographic masks and the opportunity to purchase a full wafer of robots. Both effects substantially lower the cost per area, making 0.01 USD a realistic estimate for the price of a robot as a commercial product.

### Statistical analysis

Throughout the paper, we plot data as individual points, presenting the spread within the figure; datasets are included with the publication. Error bars represent statistical ranges (interquartile, minimum/maximum) or systematic experimental error, with the specific case denoted in the figure caption.

### Supplementary Materials

#### The PDF file includes:

Methods

Figs. S1 to S9

Tables S1 to S4

Legends for movies S1 to S7

#### Other Supplementary Material for this manuscript includes the following:

Movies S1 to S7

MDAR Reproducibility Checklist

### REFERENCES AND NOTES

- A. M. Flynn, "Gnat robots (and how they will change robotics)" (Working paper, MIT Artificial Intelligence Laboratory, 1987); <https://dspace.mit.edu/handle/1721.1/41184>.
- W. S. N. Trimmer, Microrobots and micromechanical systems. *Sens. Actuators* **19**, 267–287 (1989).
- R. S. Pierre, S. Bergbreiter, Toward autonomy in sub-gram terrestrial robots. *Annu. Rev. Control Robot. Auton. Syst.* **2**, 231–252 (2019).
- S. Hollar, A. Flynn, C. Bellew, K. S. J. Pister, "Solar powered 10 mg silicon robot," in *IEEE The Sixteenth Annual International Conference on Micro Electro Mechanical Systems, 2003. MEMS-03 Kyoto* (IEEE, 2003), pp. 706–711.
- Y. Lee, G. Chen, S. Hanson, D. Sylvester, D. Blaauw, "Ultra-low power circuit techniques for a new class of sub-mm<sup>3</sup> sensor nodes," in *IEEE Custom Integrated Circuits Conference 2010* (IEEE, 2010), pp. 1–8.
- M. Zhu, O. G. Schmidt, Batteries for small-scale robotics. *MRS Bull.* **49**, 115–124 (2024).
- A. Singer, J. T. Robinson, Wireless power delivery techniques for miniature implantable bioelectronics. *Adv. Healthc. Mater.* **10**, 2100664 (2021).
- S. Bergbreiter, "Effective and efficient locomotion for millimeter-sized microrobots," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2008), pp. 4030–4035. <https://ieeexplore.ieee.org/document/4651167>.
- S. Palagi, P. Fischer, Bioinspired microrobots. *Nat. Rev. Mater.* **3**, 113–124 (2018).
- M. Sitti, *Mobile Microrobotics* (MIT Press, 2017).
- S. Chowdhury, W. Jing, D. J. Cappelleri, Controlling multiple microrobots: Recent progress and future challenges. *J. Micro Bio Robot.* **10**, 1–11 (2015).
- F. Soto, E. Karshalev, F. Zhang, B. E. Fernandez de Avila, A. Nourhani, J. Wang, Smart materials for microrobots. *Chem. Rev.* **122**, 5365–5403 (2022).
- T.-Y. Huang, H. Gu, B. J. Nelson, Increasingly intelligent micromachines. *Annu. Rev. Control Robot. Auton. Syst.* **5**, 279–310 (2022).
- M. F. Reynolds, A. J. Cortese, Q. Liu, Z. Zheng, W. Wang, S. L. Norris, S. Lee, M. Z. Miskin, A. C. Molnar, I. Cohen, P. L. McEuen, Microscopic robots with onboard digital control. *Sci. Robot.* **7**, eabq2296 (2022).
- W. A. Churaman, L. J. Currano, C. J. Morris, J. E. Rajkowski, S. Bergbreiter, The first launch of an autonomous thrust-driven microrobot using nanoporous energetic silicon. *J. Microelectromech. Syst.* **21**, 198–205 (2011).
- X. Wu, I. Lee, Q. Dong, K. Yang, D. Kim, J. Wang, Y. Peng, Y. Zhang, M. Saligane, M. Yasuda, K. Kumeno, F. Ohno, S. Miyoshi, M. Kawaminami, D. Sylvester, D. Blaauw, "A 0.04mm<sup>3</sup> 16nW wireless and batteryless sensor system with integrated cortex-M0+ processor and optical communication for cellular temperature measurement," in *2018 IEEE Symposium on VLSI Circuits* (IEEE, 2018), pp. 191–192.
- D. Chimene, K. M. K. Queener, B. S. Ko, M. McShane, M. Daniele, Insertable biosensors: Combining implanted sensing materials with wearable monitors. *Annu. Rev. Biomed. Eng.* **26**, 197–221 (2024).
- C. Shi, V. Andino-Pavlovsky, S. A. Lee, T. Costa, J. Elloian, E. E. Konofagou, K. L. Shepard, Application of a sub-0.1-mm<sup>3</sup> implantable mote for in vivo real-time wireless temperature sensing. *Sci. Adv.* **7**, eabf6312 (2021).
- L. Xu, M. Lassiter, X. Wu, Y. Kim, J. Lee, M. Yasuda, M. Kawaminami, M. Miskin, D. Blaauw, D. Sylvester, "A 210 × 340 × 50 μm integrated CMOS system for micro-robots with energy harvesting, sensing, processing, communication, and actuation," in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, 2022), pp. 1–3. <https://doi.org/10.1109/ISSCC42614.2022.9731743>.
- M. F. Reynolds, M. Z. Miskin, Materials for electronically controllable microactuators. *MRS Bull.* **49**, 107–114 (2024).
- G. B. West, W. H. Woodruff, J. H. Brown, Allometric scaling of metabolic rate from molecules and mitochondria to cells and mammals. *Proc. Natl. Acad. Sci. U.S.A.* **99**, 2473–2478 (2002).
- B. Wang, K. Kostarelos, B. J. Nelson, L. Zhang, Trends in micro-/nanorobotics: Materials development, actuation, localization, and system integration for biomedical applications. *Adv. Mater.* **33**, 2002047 (2021).
- M. Z. Miskin, A. J. Cortese, K. Dorsey, E. P. Esposito, M. F. Reynolds, Q. Liu, M. Cao, D. A. Muller, P. L. McEuen, I. Cohen, Electronically integrated, mass-manufactured, microscopic robots. *Nature* **584**, 557–561 (2020).
- Q. Liu, W. Wang, M. F. Reynolds, M. C. Cao, M. Z. Miskin, T. A. Arias, D. A. Muller, P. L. McEuen, I. Cohen, Micrometer-sized electrically programmable shape-memory actuators for low-power microrobotics. *Sci. Robot.* **6**, eabe6663 (2021).
- L. C. Hanson, W. H. Reinhardt, S. Shrager, T. Sivakumar, M. Z. Miskin, Electrokinetic propulsion for electronically integrated microscopic robots. *Proc. Natl. Acad. Sci. U.S.A.* **122**, e2500526122 (2025).
- C. Shi, T. Costa, J. Elloian, Y. Zhang, K. L. Shepard, A 0.065-mm<sup>3</sup> monolithically-integrated ultrasonic wireless sensing mote for real-time physiological temperature monitoring. *IEEE Trans. Biomed. Circuits Syst.* **14**, 412–424 (2020).
- S. Jeong, Z. Foo, Y. Lee, J.-Y. Sim, D. Blaauw, D. Sylvester, A fully-integrated 71 nW CMOS temperature sensor for low power wireless sensor nodes. *IEEE J. Solid State Circuits* **49**, 1682–1693 (2014).
- Y.-S. Lin, D. Sylvester, D. Blaauw, "An ultra low power 1V, 220nW temperature sensor for passive wireless applications," in *2008 IEEE Custom Integrated Circuits Conference* (IEEE, 2008), pp. 507–510.
- Z. Shenghua, W. Nanjian, "A novel ultra low power temperature sensor for UHF RFID tag chip," in *2007 IEEE Asian Solid-State Circuits Conference* (IEEE, 2007), pp. 464–467.
- M. K. Law, A. Bermak, A 405-nW CMOS temperature sensor based on linear MOS operation. *IEEE Trans. Circuits Syst. II: Express Briefs* **56**, 891–895 (2009).
- A. Vaz, A. Ubarretxena, I. Zalvide, D. Pardo, H. Solar, A. Garcia-Alonso, R. Berenguer, Full passive UHF tag with a temperature sensor suitable for human body temperature monitoring. *IEEE Trans. Circuits Syst. II: Express Briefs* **57**, 95–99 (2010).
- K. Souri, Y. Chae, F. Thus, K. Makinwa, "12.7 A 0.85V 600nW all-CMOS temperature sensor with an inaccuracy of ±0.4°C (3σ) from -40 to 125°C," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (IEEE, 2014), pp. 222–223.
- P. Chen, C.-C. Chen, C.-C. Tsai, W.-F. Lu, A time-to-digital-converter-based CMOS smart temperature sensor. *IEEE J. Solid State Circuits* **40**, 1642–1648 (2005).
- P. Chen, T.-K. Chen, Y.-S. Wang, C.-C. Chen, A time-domain sub-micro watt temperature sensor with digital set-point programming. *IEEE Sens. J.* **9**, 1639–1646 (2009).
- D. Shim, H. Jeong, H. Lee, J. Rhee, D.-K. Jeong, S. Kim, A process-variation-tolerant on-chip CMOS thermometer for auto temperature compensated self-refresh of low-power mobile DRAM. *IEEE J. Solid State Circuits* **48**, 2550–2557 (2013).
- C.-K. Wu, W.-S. Chan, T.-H. Lin, "A 80kS/s 36μW resistor-based temperature sensor using BGR-free SAR ADC with a unevenly-weighted resistor string in 0.18μm CMOS," in *2011 Symposium on VLSI Circuits - Digest of Technical Papers* (IEEE, 2011), pp. 222–223.
- S. Hwang, J. Koo, K. Kim, H. Lee, C. Kim, A 0.008 mm<sup>2</sup> 500 sp/s 469 kS/s Frequency-to-digital converter based CMOS temperature sensor with process variation compensation. *IEEE Trans. Circuits Syst. I: Regular Papers* **60**, 2241–2248 (2013).
- K. Souri, Y. Chae, K. A. A. Makinwa, A CMOS temperature sensor with a voltage-calibrated inaccuracy of ±0.15°C (3σ) from -55°C to 125°C. *IEEE J. Solid State Circuits* **48**, 292–301 (2013).
- Q. Zhang, W. Duan, T. Edwards, T. Ansell, D. Blaauw, D. Sylvester, M. Saligane, An open-source and autonomous temperature sensor generator verified with 64 instances in SkyWater 130 nm for comprehensive design space exploration. *IEEE Solid State Circuits Lett.* **5**, 174–177 (2022).
- T. Bai, N. Gu, Micro/nanoscale thermometry for cellular thermal sensing. *Small* **12**, 4590–4610 (2016).
- M. You, C. Chen, L. Xu, F. Mou, J. Guan, Intelligent micro/nanomotors with taxis. *Acc. Chem. Res.* **51**, 3006–3014 (2018).

42. H. Zhou, C. C. Mayorga-Martinez, S. Pané, L. Zhang, M. Pumera, Magnetically driven micro and nanorobots. *Chem. Rev.* **121**, 4999–5041 (2021).
43. Y. Zhou, L. Dai, N. Jiao, Review of bubble applications in microrobotics: Propulsion, manipulation, and assembly. *Micromachines* **13**, 1068 (2022).
44. L. Ren, W. Wang, T. E. Mallouk, Two forces are better than one: Combining chemical and acoustic propulsion for enhanced micromotor functionality. *Acc. Chem. Res.* **51**, 1948–1956 (2018).
45. Y. Alapan, O. Yasa, B. Yigit, I. C. Yasa, P. Erkok, M. Sitti, Microrobotics and microorganisms: Biohybrid autonomous cellular robots. *Annu. Rev. Control Robot. Auton. Syst.* **2**, 205–230 (2019).
46. G. Wang, T. V. Phan, S. Li, J. Wang, Y. Peng, G. Chen, J. Qu, D. I. Goldman, S. A. Levin, K. Pienta, S. Amend, R. H. Austin, L. Liu, Robots as models of evolving systems. *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2120019119 (2022).
47. J. Aguilar, T. Zhang, F. Qian, M. Kingsbury, B. McInroe, N. Mazouchova, C. Li, R. Maladen, C. Gong, M. Travers, R. L. Hatton, H. Choset, P. B. Umbanhowar, D. I. Goldman, A review on locomotion robophysics: The study of movement at the intersection of robotics, soft matter and dynamical systems. *Rep. Prog. Phys.* **79**, 110001 (2016).
48. D. Cappelleri, D. Efthymiou, A. Goswami, N. Vitoroulis, M. Zavlanos, Towards mobile microrobot swarms for additive micromanufacturing. *Int. J. Adv. Robot. Syst.* **11**, 150 (2014).
49. J. Li, B. Esteban-Fernández De Ávila, W. Gao, L. Zhang, J. Wang, Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification. *Sci. Robot.* **2**, eaam6431 (2017).
50. K. Yang, Q. Dong, W. Jung, Y. Zhang, M. Choi, D. Blaauw, D. Sylvester, “9.2 A 0.6nJ  $-0.22/+0.19^{\circ}\text{C}$  inaccuracy temperature sensor using exponential subthreshold oscillation dependence,” in *2017 IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, 2017), pp. 160–161.
51. W. Lim, T. Jang, I. Lee, H.-S. Kim, D. Sylvester, D. Blaauw, “A 380pW dual mode optical wake-up receiver with ambient noise cancellation,” in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)* (IEEE, 2016), pp. 1–2.

**Acknowledgments:** We thank M. Reynolds and D. Koditschek for helpful discussions and M. Azadi and the Singh Center for Nanotechnology staff for technical support during the fabrication process. **Funding:** This work was supported by the National Science Foundation (NSF 2221576), the University of Pennsylvania Office of the President, the Air Force Office of Scientific Research (AFOSR FA9550-21-1-0313), the Army Research Office (ARO YIP W911NF-17-S-0002), the Packard Foundation, the Sloan Foundation, and the NSF National Nanotechnology Coordinated Infrastructure Program (NNCI-2025608), which supports the Singh Center for Nanotechnology. **Author contributions:** D.B. and M.Z.M. conceptualized the work and administered the project. **Methodology:** M.M.L., J.L., L.H., W.H.R., L.X., M.Z.M., D.B., and D.S. contributed to the methodology. Experiments were carried out by M.M.L., J.L., L.X., and K.S. Visualizations were created by M.M.L., M.Z.M., and J.L. D.B., M.Z.M., and M.Y. acquired funding and supervised the work. M.M.L. and M.Z.M. wrote the manuscript with all authors contributing. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to support the conclusions of this manuscript are included in the main text or Supplementary Materials or Dryad repository (DOI: 10.5061/dryad.x0k6djhxt).

Submitted 10 February 2025  
Accepted 13 November 2025  
Published 10 December 2025  
10.1126/scirobotics.adu8009

## Microscopic robots that sense, think, act, and compute

Maya M. Lassiter, Jungho Lee, Kyle Skelil, Li Xu, Lucas Hanson, William H. Reinhardt, Dennis Sylvester, Mark Yim, David Blaauw, and Marc Z. Miskin

*Sci. Robot.* **10** (109), eadu8009. DOI: 10.1126/scirobotics.adu8009

### View the article online

<https://www.science.org/doi/10.1126/scirobotics.adu8009>

### Permissions

<https://www.science.org/help/reprints-and-permissions>

Use of this article is subject to the [Terms of service](#)

---

*Science Robotics* (ISSN 2470-9476) is published by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.

Copyright © 2025 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works