nature electronics

SUPPLEMENTARY INFORMATION

In the format provided by the authors and unedited.

The first family of application-specific integrated circuits for programmable metasurfaces

Loukas Petrou [1]✉, Kypros M. Kossifos [1]✉, Marco A. Antoniades [1,2] and Julius Georgiou [1]

[1] Department of Electrical and Computer Engineering, University of Cyprus ✉e-mail: petrou.loukas@ucy.ac.cy; kossifos.kypros@ucy.ac.cy
[2] Department of Electrical, Computer and Biomedical Engineering, Ryerson University, Toronto, Canada

Supplementary Information

Supplementary Note 1. Loading Element Measurements

The family of ASICs operates at the sub-6GHz band with four versions that are optimized for 5 GHz applications (Designs 1, 2, 5 and 6) and two designs for 3 GHz applications (Designs 3 and 4). The designs also possess two chip footprint versions. The chip footprint for Designs 1 and 2 can be seen in Fig. 1a. This footprint has a separate ground for the control circuit (DGND) and the RF ground (AGND). For these two designs an RF choke can be added in path of the RF ground of the ASIC and the PCB's ground. This is particularly useful when there is a need to have a floating loading element. Designs 3 to 6 pin map, as shown in Fig. 1b, have digital and RF common ground and each loading element has three ground pins near it to minimize the parasitics within the ASIC that are introduced by metallic lines.

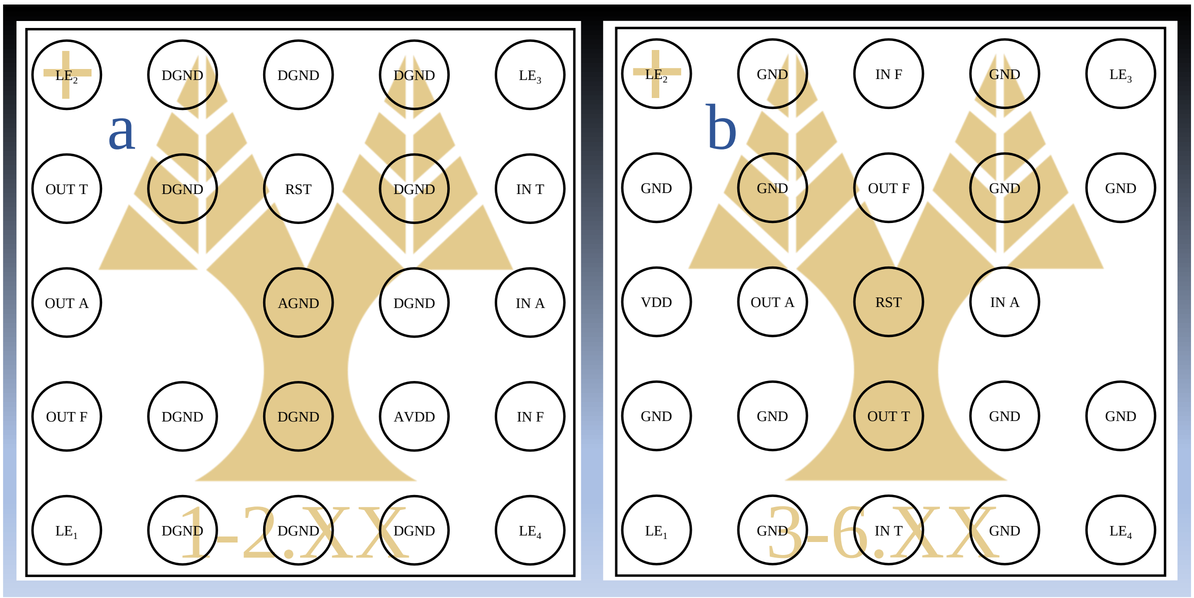


Fig. 1 | Chip footprints of the family of ASICs for the view of the logo facing up. a, Footprint for designs 1 and 2 and **b,** Footprint for designs 3 to 6.

Table 1| Pin description

Pin Name	Description
LE$_x$	Loading Element x terminal
IN T	Input True Signal
IN A	Input Ack Signal
IN F	Input False Signal
OUT T	Output True Signal
OUT A	Output Ack Signal
OUT F	Output False Signal
DVDD	Digital Power
AVDD	Analogue Power
DGND	Digital Ground
AGND	RF Ground
GND	Common Digital and RF Ground
RST	Negative Trigger Reset (or $\overline{\text{RST}}$)

A description of various pins is included in Table 1. The ASICs and the measurement set-up can be seen in Fig. 2. The ASICs were populated on PCBs with a high frequency substrate (Rogers RO4350) and 50-ohm grounded coplanar waveguide (CPWG) transmission lines connect the loading elements to the coaxial connectors at the edge of the PCB. A Keysight N5227B four port VNA was used to acquire the scattering parameters in touchstone file format. A MATLAB script was used to control the VNA and simultaneously program the ASIC through a field programmable gate array Opal Kelly XEM6010. The measurements were performed at room temperature. The touchstone files are de-embedded with the through-reflect-line technique. The de-embedded measured data can be seen in Fig. 3 for all six the designs. In the same figure, for all the designs the perimeter of the loading element range can be seen in dashed red at the design frequency and sample points are plotted with black dots inside the perimeter. Furthermore, measurements for the four corners of the range are plotted from 2 to 6GHz.

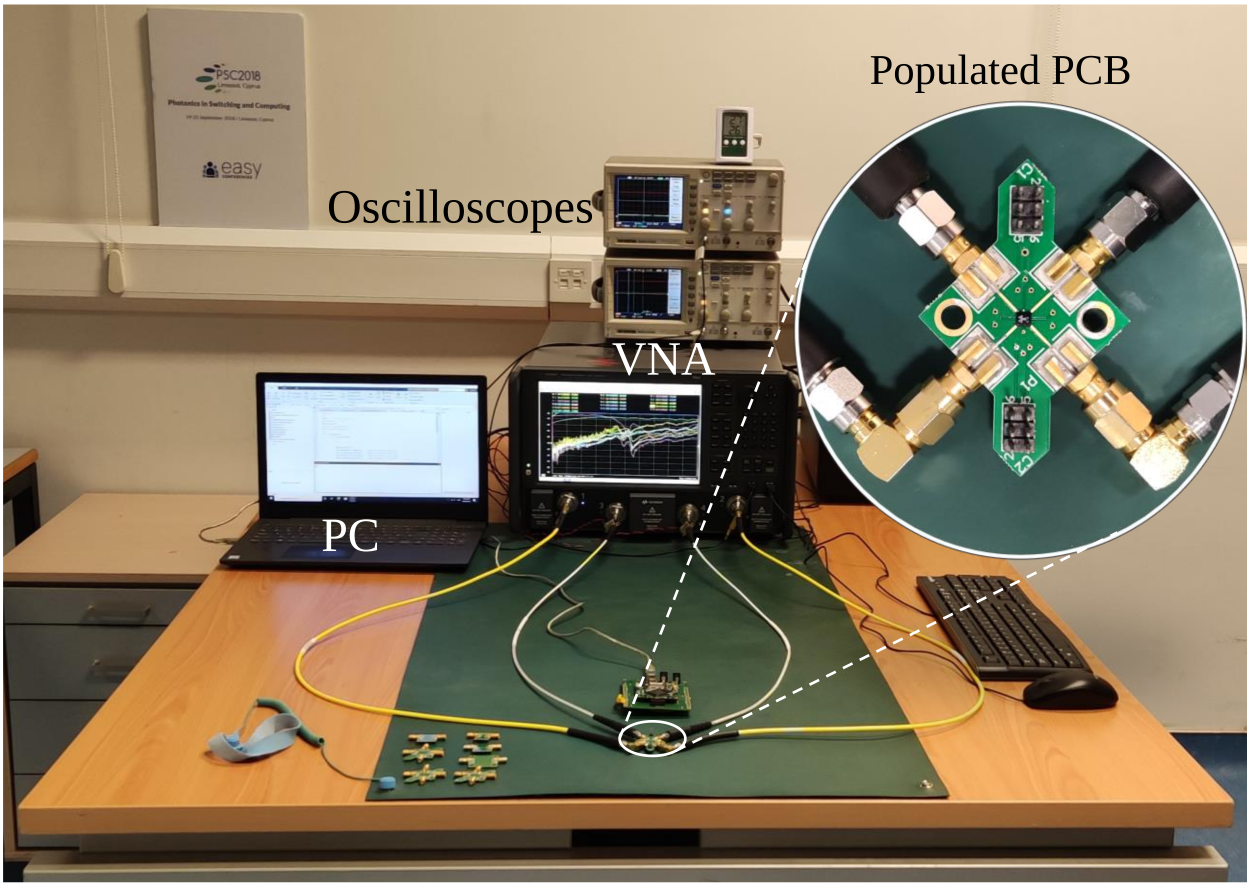


Fig. 2| Loading element measurement setup.

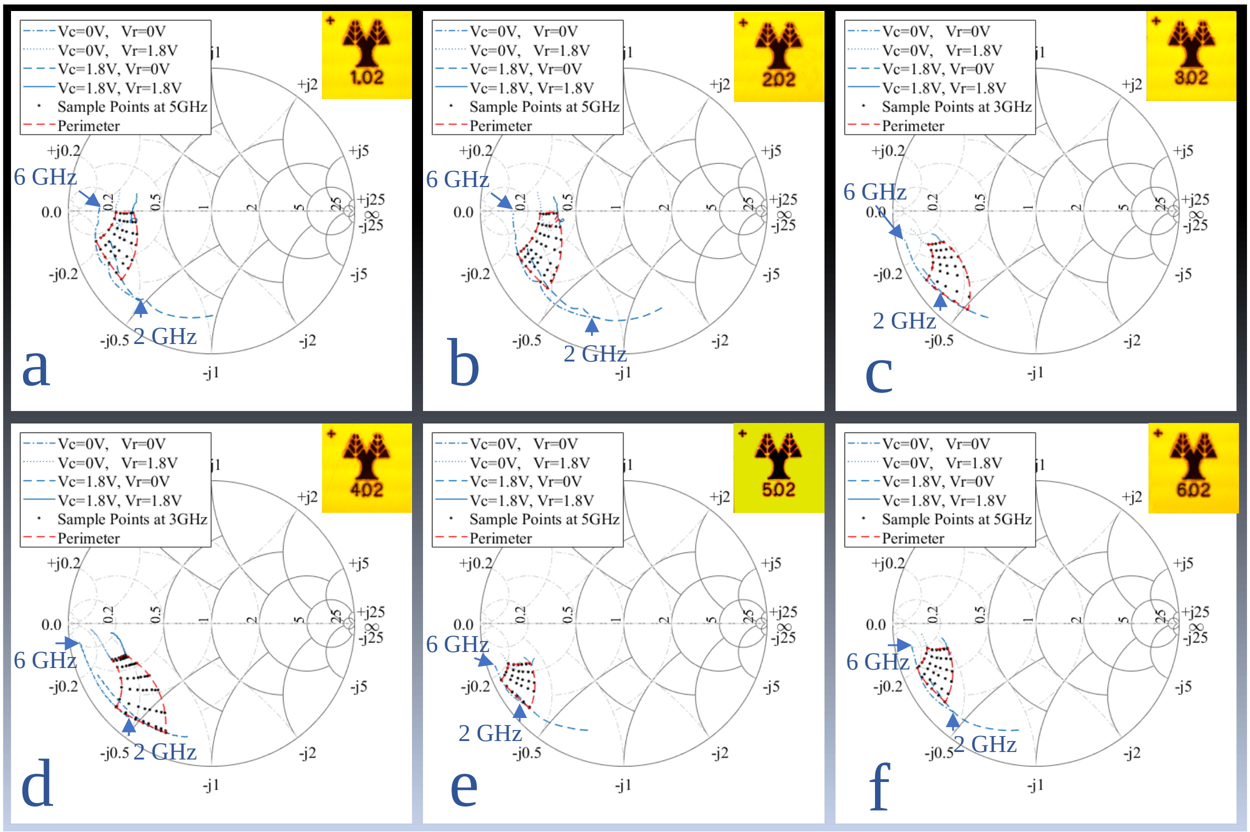
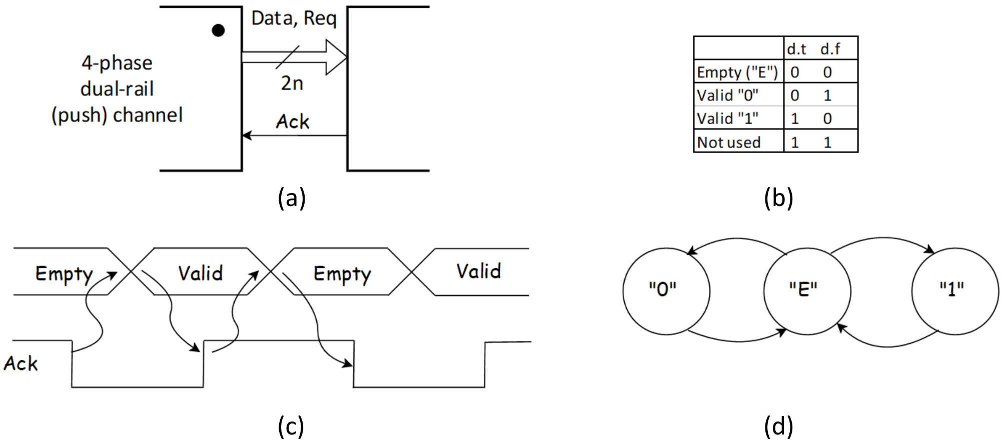


Fig. 3| Measured loading element impedance ranges.a, Design 1 **b,** Design 2 **c,** Design 3 **d,** Design 4 **e,** Design 5 and **f,** Design 6.

Supplementary Note 2. 4-phase dual-rail protocol

The 4-phase dual-rail protocol is an asynchronous digital communication protocol for transmitting/receiving data without the use of external clock signal. Instead, it uses handshaking, using the well-known ‘request’ and ‘acknowledge’ signal-wires. It is a robust protocol due to its delay insensitivity to gate-delays as opposed to other typical asynchronous communication protocols. In Fig. 4, a graphical interpretation of the protocol is illustrated. Fig. 4a shows the signals exchanged at the channel. We use two wires for every bit that we transmit thus the data bus requires 2n wires (n represents the number of bits that we transmit in parallel). One wire of a bit is the ‘data.true’ or ‘d.t’ and used for signal logic ‘1’ and the other wire is ‘data.false’ or ‘d.f’ and used for signal logic ‘0’. The request signal is encoded into the data signals and is set to ‘1’ only when all data signals are valid. A valid bit is considered when one of the two wires is logic ‘1’ as shown in the table of Fig. 4b. In Fig. 4c, the signal transitions in the channel is shown and the procedure is as follows:

Starting the communication, all signals are ‘0’ which means we have ‘Empty’ data and the ‘Ack’ signal is low. Then, the sender issues valid data as shown in Fig. 4b. The receiver stores the valid data and acknowledges this, through ‘Ack’ = ‘1’. The data then becomes ‘Empty’ and ‘Ack’ becomes ‘0’. At this point, the sender and the receiver have exchanged one bit of information and the next cycle is ready to begin. A simpler representation of the communication is shown in Fig. 4d, which only shows the state transition. Notice that whatever the data is, the state must return to ‘Empty’ before issuing new data. This is known as ‘return-to-zero’ or ‘RTZ’ line coding.



	d.t	d.f
Empty ("E")	0	0
Valid "0"	0	1
Valid "1"	1	0
Not used	1	1

Fig. 4| The 4-phase dual-rail protocol. a, Communication channel between two consecutive nodes **b,** Data representation **c,** Signal transition for bit-to-bit communication and **d,** State Diagram for bit-to-bit communication.

Fig. 5 shows the timing diagram of the 4-phase dual-rail protocol. Time starts at the top of the diagram and increases going downwards. The diagram assumes one bit is exchanged thus the total number of wires for the communication are three (data.t, data.f, and ack).

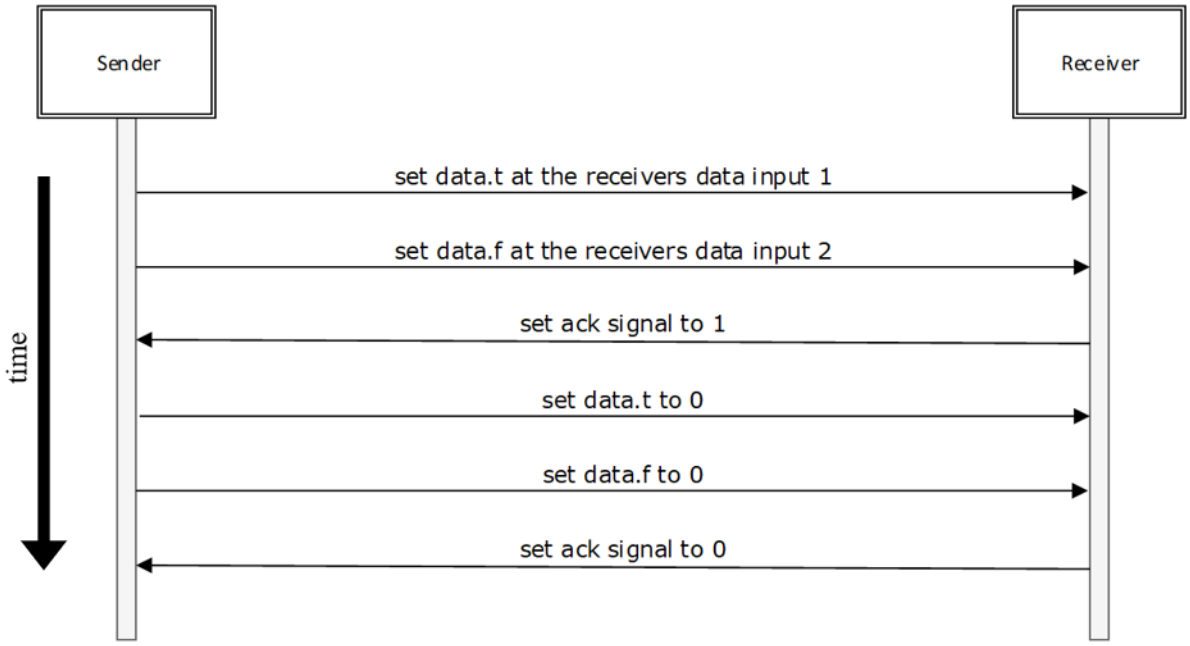


Fig. 5| Timing diagram of one communication cycle between a sender node and a receiver node.

Fig. 6 illustrates the channel between two consecutive control circuits. A configuration packet consists of 64 bits and requires an average of 1 µs to be transmitted from the sender to the receiver. This implies that the frequency for sending packets is 1 MHz. The bit rate of each chip is measured at 68 Mbits/s. The energy consumed per bit is 79 pJ and the energy consumed per packet is 5.1 nJ. The energy consumption includes the memory cells and the output buffers of the chip. Fig. 6 also shows the signal transition for the binary sequence ‘0101’ which implements the 4-phase dual-rail protocol. The data rises first (‘data.true’ or ‘data.false’ = ‘1’), followed by a rise in the acknowledgment (‘data.ack’ = ‘1’) after the data has been stored at the receiver’s memory cell. To end the cycle, the data return to its starting state (‘0’) followed by the acknowledge returning to its starting state (‘0’).

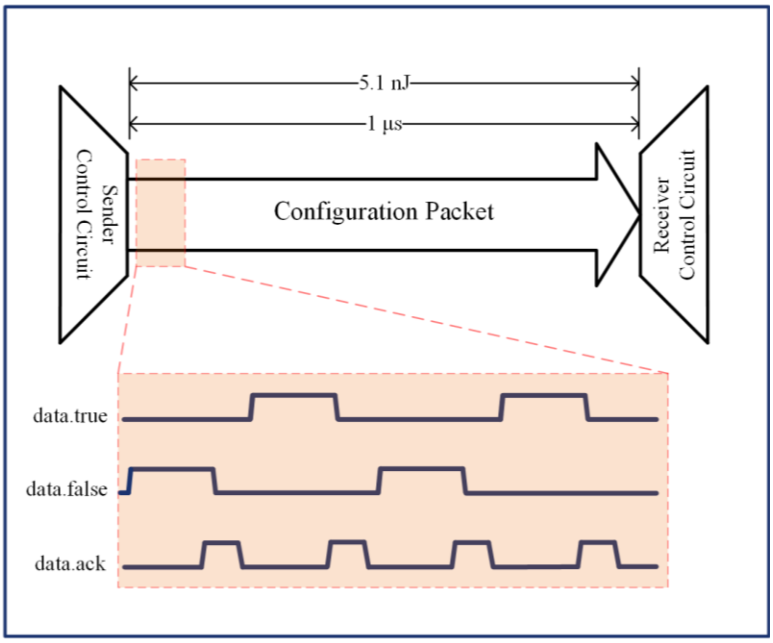


Fig. 6| The channel between a sender control circuit and a receiver control circuit transmitting one configuration packet. The figure zooms on the transmission of the binary sequence ‘0101’.

Supplementary Note 3. Graphical user Interface (GUI) for testing the communication performance.

Two programs were created for testing the chips, one for manual operation and one for automatic operation. For each of these, a graphical user interface (GUI) has been created to visualize the response of the chips.

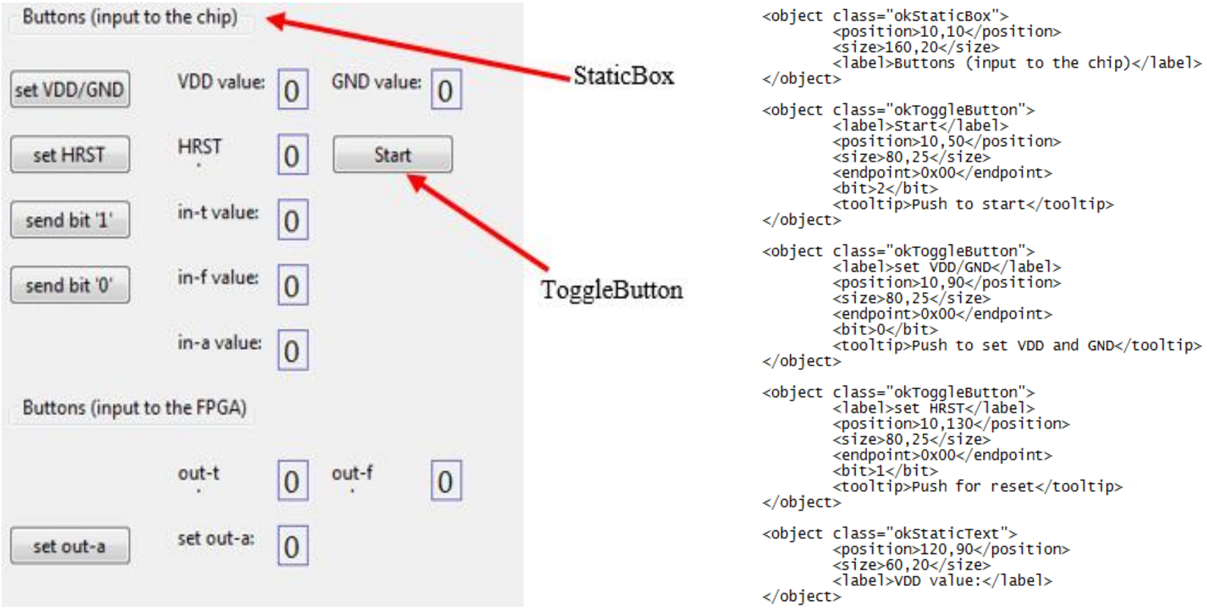
A. Manual Operation of Chip

The GUI for the manual operation is shown in Fig. 7. Fig. 7a shows the GUI, and Fig. 7b shows part of the .xml file that creates the objects on the interface. For example, the text at the top of the GUI, ‘Buttons (input to the chip)’ is of type ‘StaticBox’ and in the .xml file its position, size, and label can be defined.

In this program, the user can manually set the value of each input (to the chip) signal. The output (of the chip) signals are set by the chip and the user can visualize this response on the GUI. The user must manually execute the dual-rail protocol in order to communicate with the chip and provide it with packets. Below (Table 2), we describe each box of the GUI.

Table 2| Buttons and Digits for the manual operation GUI.

	Buttons
Start	Start the program
Set VDD/GND	Make VDD value = ‘1’ and GND = ‘0’
Set HRST	Make Hard Reset = ‘1’ (Negative Triggered)
Send bit ‘1’	Make In-t value = ‘1’ and In-f value = ‘0’
Send bit ‘0’	Make In-f value = ‘1’ and In-t value = ‘0’
Set out-a	Make Out-a value = ‘1’
	Digits
In-a value	Acknowledgment value from chip
Out-t value	Output true value from chip
Out-f value	Output false value from chip



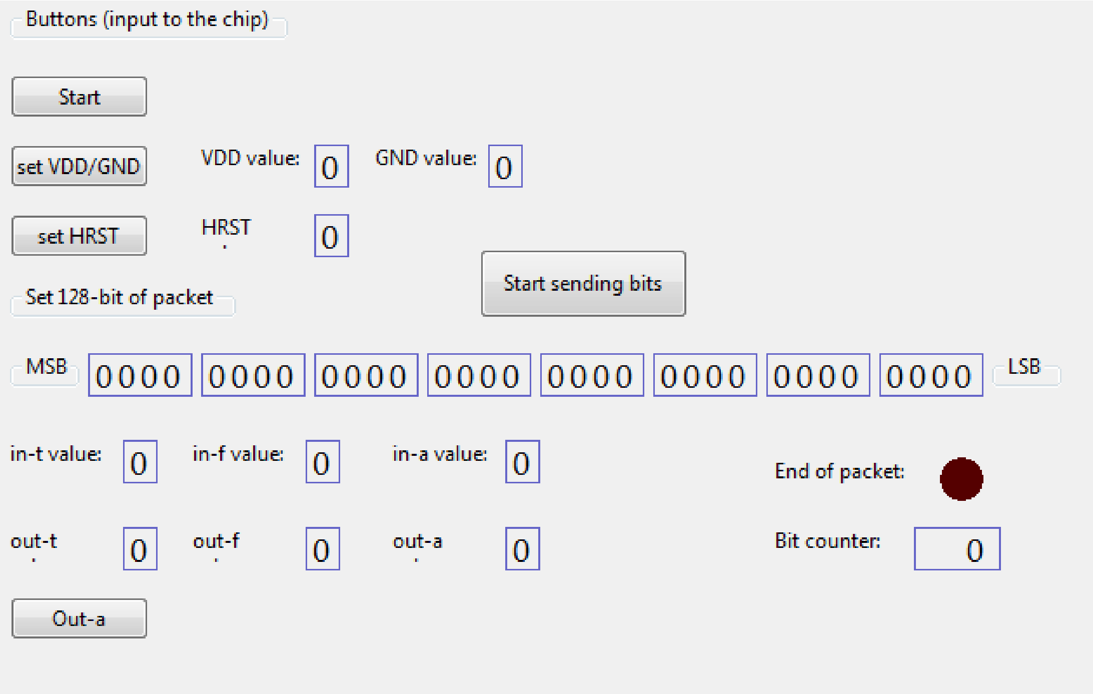
```
<object class="okStaticBox">
        <position>10,10</position>
        <size>160,20</size>
        <label>Buttons (input to the chip)</label>
</object>

<object class="okToggleButton">
        <label>Start</label>
        <position>10,50</position>
        <size>80,25</size>
        <endpoint>0x00</endpoint>
        <bit>2</bit>
        <tooltip>Push to start</tooltip>
</object>

<object class="okToggleButton">
        <label>set VDD/GND</label>
        <position>10,90</position>
        <size>80,25</size>
        <endpoint>0x00</endpoint>
        <bit>0</bit>
        <tooltip>Push to set VDD and GND</tooltip>
</object>

<object class="okToggleButton">
        <label>set HRST</label>
        <position>10,130</position>
        <size>80,25</size>
        <endpoint>0x00</endpoint>
        <bit>1</bit>
        <tooltip>Push for reset</tooltip>
</object>

<object class="okStaticText">
        <position>120,90</position>
        <size>60,20</size>
        <label>VDD value:</label>
</object>
```

(a) (b)

Fig. 7| a, Graphical User Interface for the manual program and **b,** Xml file that created the objects in **a.**

B. Automatic Operation of Chip

The automatic program respects the dual-rail protocol and sends multiple bits serially to the chips. Specifically, the program sends to the chip 128 bits that are chosen by the user in hexadecimal (HEX) form, entered through the GUI (Fig. 8). At the beginning of the program, the user specifies the packet in the boxes under the ‘Set 128-bit of packet’ text. The bits are sent starting from the MSB and ending at the LSB. The program also keeps track of the bits sent (‘Bit counter’) and informs the user when the packets is successfully sent (‘End of packet’ flashes red).



VDD value: 0 GND value: 0

HRST 0

MSB 0000 0000 0000 0000 0000 0000 0000 0000 LSB

in-t value: 0 in-f value: 0 in-a value: 0

End of packet: ●

out-t 0 out-f 0 out-a 0

Bit counter: 0

Fig. 8| Graphical User Interface for the Automatic Operation of the Chip